Abstract Rule Based Pattern Learning with Neural Networks

Radha Manisha Kopparti
(radha.kopparti@city.ac.uk)

August 5, 2019

1 Outline of the article

The ability to learn abstractions and generalise is seen as the essence of human intelligence.[7] Since 1950s, there have been efforts to build systems that learn and think like humans.[16] It is observed that humans including infants tend to have good generalisation power when compared to the machine learning models in which hypothesis is usually approximated and may be prone to errors.

The examples proposed by Marcus[19,18,17] such as the failure to generalise equality, distinguish between even to odd numbers or the recognition of ABA or ABB patterns of syllables have attracted a significant amount of attention in psychology, particularly in the study of human language learning, but they have not been addressed systematically as problems of machine learning and neural networks.

In this article, the problem of learning abstract rules using neural networks is explained and a solution called **'Relation Based Patterns'** (RBP) which model abstract relationships based on equality is proposed. RBP creates an inductive bias in the neural networks that leads to the learning of generalisable solutions. It is observed that integration of RBP leads to almost perfect generalisation in abstract rule learning tasks with synthetic data and to improvements in neural language modelling on real-world data.

The outline of the article is as follows : introduction to the problem is briefly described followed by a section on what is abstract pattern (rule) learning, the need for inductive bias and various ways of adding inductive bias into neural networks. The RBP method and its integration along with the experiments on the tasks of abstract rule learning, character prediction and melody prediction are summarized followed by conclusions and future work.

2 Introduction

Despite the successes achieved with deep neural networks over recent years, there has been an increasing awareness that there are tasks that still elude neural network learning, specifically the generalisation from patterns to rules. Generally, humans are very effective at extracting abstract relations (eg: as in Figure 1) from sensory input, often after very brief exposure. In rule based grammar learning tasks, participants are asked to classify the input or predict the next letter after getting exposed to input stimuli generated from a random alphabet.[13,9]

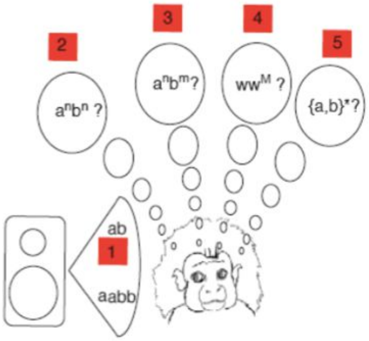


Figure 1: How do humans perceive grammatical rules?

In a famous study, Marcus et al.[19] showed that 7-month old infants learnt to recognize patterns defined by simple grammatical rules, specifically sequences of the structure ABA or ABB. They learnt the sequences from a small number of examples in just two minutes of familiarization which is evidence for the hypothesis that humans have innate understanding of identity rules. When tried to reproduce the same experiment using neural networks, the networks failed to significantly identify these abstract patterns. More specifically, it is found that feed-forward and recurrent neural networks (RNN) and their gated variants (LSTM and GRU) in standard set-ups clearly fail to learn general identity rules when presented as classification and prediction tasks. Therefore, in this work, the problem of learning abstract rules in neural networks is tackled by introducing Relation Based Patterns (RBP) as an inductive bias to model equality relationships of patterns based on grammatical rules.

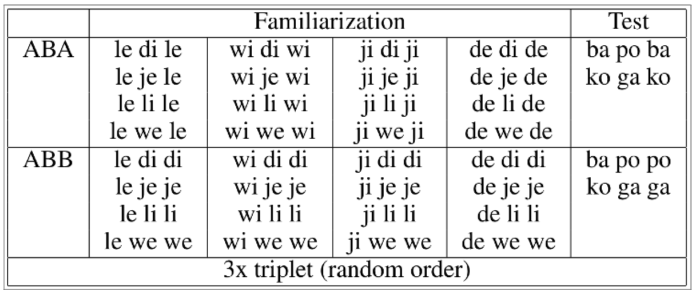
RBP model has been designed as a set of additional neurons with a rectified difference activation and fixed-weight connections that connect them to standard networks. The main idea is to model equality relations and abstract patterns as a foundation for more complex logic and systematicity, e.g. the application of grammar in natural language processing.[15,20] It is observed experimentally that on synthetic data, neural networks with suitable RBP structures learn the relevant rules and generalise with perfect classification (for abstract rule learning task) and prediction (for language modelling tasks). There are different ways of modelling relational inductive biases within the neural network architectures and here one such way is described.

The primary focus of the article is on the following aspects :

- Why can't standard neural networks learn grammatical patterns for example say based on equality?
- How can we make the neural networks learn these grammatical abstractions?
- What does it mean to other real world tasks like language modelling where there are higher level of abstractions in data?

3 What is abstract pattern (rule) learning?

In general, abstract pattern learning task comes under the popular paradigm of study within cognitive science and linguistics called as artificial grammar learning. In artificial grammar learning, the key idea is to understand human language learning by testing the ability of humans in learning artificial grammar patterns. There are two phases in the task, one is the training stage where the patterns are made familiar to the subjects and the ability to identify and generalise this new knowledge is tested in the second stage ie. testing phase. The testing phase typically comprises of symbols or sounds used in the training phase or transfer of the patterns to another set of symbols.



	Familiarization				Test
ABA	le di le le je le le li le le we le	wi di wi wi je wi wi li wi wi we wi	ji di ji ji je ji ji li ji ji we ji	de di de de je de de li de de we de	ba po ba ko ga ko
ABB	le di di le je je le li li le we we	wi di di wi je je wi li li wi we we	ji di di ji je je ji li li ji we we	de di di de je je de li li de we we	ba po po ko ga ga
3x triplet (random order)					

Figure 2: Empirical data used in the Abstract Grammar Learning Experiment by Marcus et al[19]

In abstract pattern learning, series of abstract patterns be it grammar like rules or sequences of data like music or language are used. One of the earliest work on abstract pattern learning was by[19,18] where the abstract grammar like rules in Figure 2 are shown to six month old infants and their task was to distinguish the grammatical rules. The infants were exposed to sequences of one of the forms ABA or ABB, e.g. 'le di le' or 'le di di', for a few minutes in the familiarisation phase. In the test phase the infants were exposed to sequences with a different vocabulary (e.g. 'ba po ba' and 'ba po po') and the results from the experiments was that the infants were able to learn the grammatical patterns within minutes of habituation. The same task when reproduced using a recurrent neural network, failed to distinguish the grammar patterns.

The work by[11,22,8,23] focused on using distributed representation of the inputs where the network was not able to learn the patterns as expected. Other works used localist representation[2,3,1,10] which suggests that an additional context or prior experience is necessary for the network to learn these identity rules. This has raised questions on whether the existing statistical and neural network models can generalize these abstract patterns or not.

An effective way of solving this problem is to introduce an inductive bias within the network structure to achieve better generalisation from fewer samples of training data. However, the question then arises about the type of inductive bias needed to improve the overall generalisation performance. To achieve better generalisation, the neural network should be able to learn the required knowledge and apply that to unseen circumstances or extend it beyond the scope of the actual problem.

4 Need for inductive bias?

The issues related to the lack of generalisation beyond the space covered by the input data can be addressed by adding an inductive bias in the learning system, but there is no general agreement about the nature or implementation of inductive biases for neural networks, e.g..[12,4] In recent years, there was a trend to remove human designed features from neural networks, and leave everything to be learned from the data.[6]

More recently, the problem of data efficiency has motivated a new look at inductive biases. Recent work has shown that designing specific biases into the structure of the learning process can address these problems, e.g. by choosing appropriate data organisation and filters in convolutional networks, as was show by,[20] or by adding suitable pre-defined connections as described in this article (full paper link of this work described in this article[25]). Although solutions like these are sometimes criticised as being 'hard-coded', there are good reasons for investigating them:

- Necessity for problem solving: both[25,20] show that the problem cannot be solved by standard architectures. Although there were some claims that the problem posed by[19] was solvable, it turned out that these claims could not be verified, or used very specific non-standard architectures and success was defined as showing significantly different reaction to the one class or patterns vs the other, while humans can easily learn and recognise them.

- No restriction to general learning: A criticism towards 'hard-coded' solutions is that they may be too specific and thus not generalise to other tasks than they were designed for. Our results in[25] show that this is not the case for 'concrete' vs 'abstract' patterns, as described by.[18]

- Effectiveness in real-world tasks: The experiments by[25] and[20] show that in standard tasks on language and music (word/note prediction, natural language inference) there are improvements when biases are added to the networks.

The last item above is evidence, that this is not just a theoretical problem, but that addressing the structures underlying these problems does address fundamental and relevant problems in neural network learning.

In many application domains, large amounts of reliable data are very difficult to obtain for ethical, practical or financial reasons. However, most applications of rules depends on the ability to recognise identity or similarity according to some criterion, and if that cannot be learned for new items, then the generalisation and applicability of the learning is severely limited. In the next section, various ways of adding inductive bias in standard neural networks is explained.

5 Ways of adding inductive biases

There are a number of ways in which inductive biases can be added to the model. An inductive bias would allow a learning algorithm to prioritize one solution over the other, independent of the observed data.[5]

First and foremost way, is adding bias with a *pre-defined network structure*. In this, a type of representation in the form of circuits or abstract structure is added to the standard network models, either combining with the input layer, hidden layer or the output layer as described in this work.

Another way is a Bayesian approach where bias can be added based on the *prior distribution* of various input features. Inductive bias can also be added as a *regularization term* or encoded into the network architecture, as shown by.[5]

Inductive bias can also be expressed as a part of the data generating process or within the network space of the solution. In fact, priors can be derived from the type of data and the *constraints* can be set on the model being used. This is adapting bias as an optimisation problem and solving the model based on the constraints.

There are also ways of *approximating functions and errors* of the network models as a form of having an inductive bias in the network model which has close connections with the data pre-processing step. Infact, the type of *network structure* can lead to different forms of inductive biases. For example, the type of bias for a recurrent network can be different from that of a convolution network.

There are a number of ways by which one can model the bias in the network models. Depending on the task and the domain, the following characteristics can be crucial in effective neural network learning. One such way of creating inductive bias for abstract pattern learning is RBP which is described below.

6 RBP method

To address the inability of neural networks to generalise rules in neural network learning, Relation Based Pattern (RBP) model is developed as a constructive solution, where the comparisons between input neurons and between tokens and the mappings to outputs are added as a predefined structure to the network. The purpose of this structure is to enable standard neural networks to learn abstract patterns based on the identity rules over tokens while retaining other learning abilities. The RBP model is based on an input that consists of multiple items, where each can be represented by a vector of input neurons. In the RBP model there are two major steps.

The first step is defining comparison units for detecting abstract patterns, called DR units, and the second step is adding the DR units to the neural network. DR units are used to compare corresponding neurons in different vectors. For the comparison we introduce differentiator-rectifier (DR) units, which calculate the absolute difference of two inputs: $f(x, y) = |x - y|$. One DR unit for every pair of corresponding input unit is created with the weights from the inputs to the DR units fixed at 1. There are three ways of adding RBP into the standard neural networks : *Early Fusion*, *Mid Fusion* and *Late Fusion*.

Comparing neurons : The input is a one-hot encoded vector of the current token along with the $n-1$ previous vectors for a given context length n. Different representations other than one-hot encoding are tested but in this article, the experiments are performed with inputs which are one-hot encoded.

The first level of DR units are DR_n units that are applied to every pair of corresponding input neurons (representing the same value) within a token representation, as shown in Figure 3.

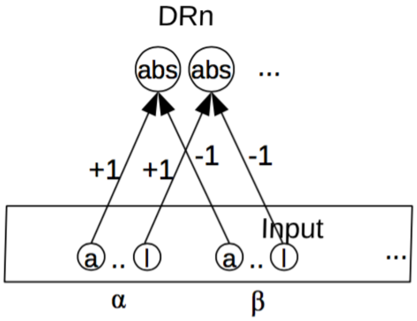
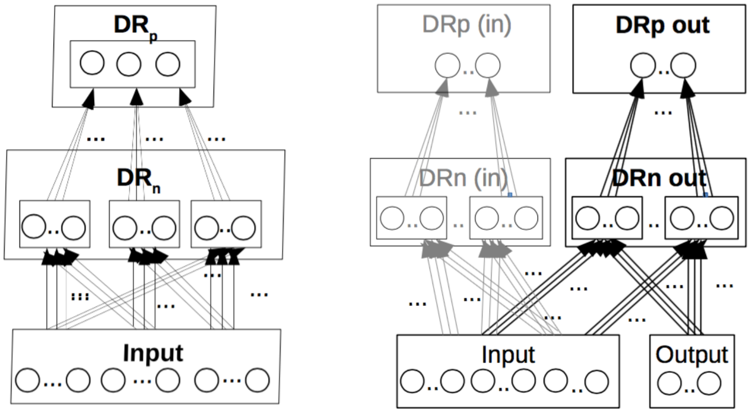


Figure 3: DR_n units comparing related inputs with an absolute of difference activation function. In one-hot encoding there are k DR_n units for every pair of input tokens, where k is the vocabulary size.

Comparing tokens : The next level of DR units are the DR_p units that sum the activations of the DR_n values that belong to one pair of tokens. Based on the sequence length n and vocabulary size a we create $k = a * n(n-1)/2$ DR_n units for all the possible pairs of tokens i.e. in the classification example, for a sequence of 3 tokens and a vocabulary size of 12, $12 * 3(3-1)/2 = 36 * 3$ DR_n units are considered. All the DR_n units for a pair of tokens are then summed in a DR_p unit using connections with a fixed weight of +1. E.g. there are $5 * (5-1)/2 = 10$ DR_p units for a context of length 5. Figure 4(a) below shows the network structure with DR_n and DR_p units.



(a) The DR_p and DR_n units that are used in the RBP structures with $3 \times k$ DR_n and 3 DR_p units for a vocabulary size k and sequence length 3.

(b) The DR_{out} structure for detecting relation between input and target. The $DR_p out$ values are calculated at training time and a model is trained to predict them conditional on $DR_p in$.

Figure 4: DR_n and DR_p units of RBP for classifcation and prediction tasks.

For the prediction case, the same approach is used to represent the difference between each input token and the next token (i.e., the target network output). In this case, n DR_p out units are created that calculate the difference between each input in the given context and the next token. There are $k * n$ DR_n out units that compare the corresponding neurons for each pair of input/output tokens, in the same way as for the pairs of input tokens. The overall network structure is shown in Figure 4(b).

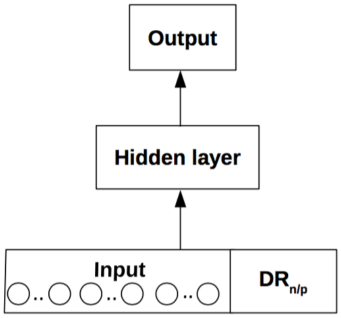


Figure 5: Overview of the RBP Early Fusion.

7 RBP Neural Network Integration

Early Fusion : In this approach, DR_n or DR_p units are added as additional inputs to the network, concatenated with the normal input. In Figure 5, the RBP_n/RBP_p structure is depicted. Early fusion is used in both the prediction and classification tasks.

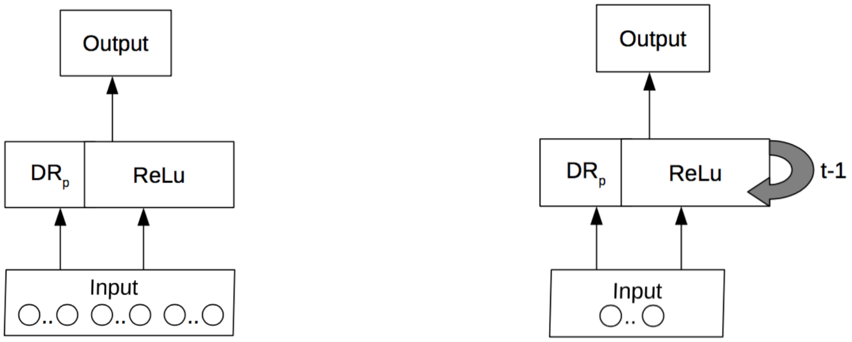


Figure 6: Overview of the RBP Mid Fusion

Mid Fusion : The DR_p units are added to the hidden layer. Figure 6 shows the mid fusion structure for the feed-forward network and recurrent network respectively. Mid Fusion approach is used for classification and prediction tasks.

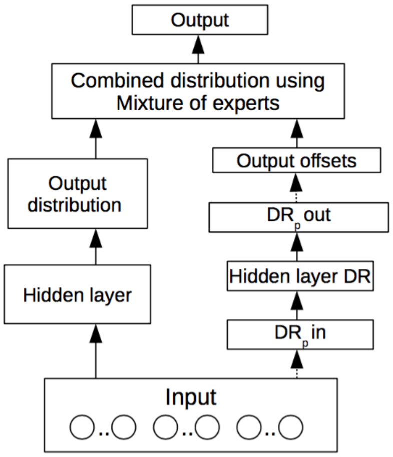


Figure 7: Overview of the RBP Late Fusion

Late Fusion : In this approach, the same structure of RBP as described above is used, in addition the probability of identity relations between the input and the output, i.e., that the token in the current context is repeated as the next token is estimated. A structure called DR_p out is used for this, and from there the output offsets are projected back to the vocabulary, to generate a probability offset for the tokens appearing in the context. Figure 7 gives an overview of the RBP late fusion scheme and for a detailed explanation of the process please refer to our paper.[25] Late fusion is only used for the prediction task.

8 Experiments and results

8.1 Learning Abstract Rules

In this task, triples of the forms ABA, ABB, ABC, AAB and BAB are given to the network as a supervised formulation of[19] with some variants as a classification task. A 75/25 train/test split with separate vocabulary between them is used. The results of the experiments are given in Table 1. It is observed that without RBP, neural networks never improve above chance level (50%), and with RBP it leads to significant improvement with almost perfect results observed for RBP in Mid Fusion.

Type	Standard FFNN	Early Fusion	Mid Fusion
1. ABA vs other	50 (1.86)	65 (1.26)	100 (0.00)
2. ABB vs other	50 (1.83)	65 (1.29)	100 (0.00)
3. ABA-BAB vs other	50 (1.73)	75 (1.22)	100 (0.05)
4. ABA vs ABB	50 (1.81)	55 (1.18)	100 (0.00)
5. ABC vs other	50 (1.68)	65 (1.04)	100 (0.00)

Table 1: Accuracy (in %) and standard deviation over 10 simulations (in brackets) using different models for Abstract Pattern Learning (ABA vs other, ABB vs other, ABA-BAB vs other, ABA vs ABB, ABC vs other).

8.2 Character and Melody Prediction

For character prediction, recurrent neural networks and their gated variants (LSTM and GRU) are used on a subset of the Gutenberg electronic book collection[1], consisting of 42252 words. The experiments are performed with 2 hidden layers with 50 neurons each, an initial learning rate of 0.01 and the network training converged after 30 epochs. A train/valid/test split of 50/25/25 was used. The results using context size 5 are summarised in Table 2 for simple network without RBP and with RBP in Early, Mid and Late Fusion for RNN, GRU and LSTM respectively. It is observed that with RBP there is consistent decrease in the overall cross entropy loss for all the models and the best performance has been observed with RBP in Late Fusion using LSTM.

[1] https://www.gutenberg.org/

Type	RNN	GRU	LSTM
Simple Network	3.8281	3.8251	3.8211
Early Fusion	3.8254	3.8163	3.8162
Mid Fusion	3.8148	3.8134	3.8112
Late Fusion	**3.8076**	**3.8053**	**3.8032**

Table 2: Average Cross Entropy Loss per predicted character for Character Prediction Task using context length 5.

In another experiment, RBP is tested on a pitch prediction task in melodies using the Essen Folk Song Collection[21] with recurrent neural networks and their gated variants (LSTM and GRU). Pitch patterns of melody have a lot of repetition cues and previous works on pitch prediction hasn't explored this aspect of modelling abstract repetition patterns in folk melodies as required. Using RBP, the abstract repetition patterns in melodies are modelled here. For the pitch prediction experiments, a grid search for hyper parameter tuning is performed, with [10,30,50,100] as the size of the hidden layer and [30,50] epochs with learning rate set to 0.01, with one hidden layer and context length of size 5. The results in Table 3 summarize the results and shows a consistent reduction in cross entropy with RBP in various forms of integrations. Similar to character prediction, the best performance is observed with LSTM combined with RBP in Late Fusion.

Type	RNN	GRU	LSTM
Simple Network	2.6994	2.6714	2.6589
Early Fusion	2.6942	2.6702	2.6564
Mid Fusion	2.6837	2.6623	2.6483
Late Fusion	**2.6713**	**2.6514**	**2.6386**

Table 3: Average Cross Entropy Loss per note for Melody Prediction Task using context length 5.

9 Key findings and Conclusions

Overall, through this study

- several neural network architectures like feed-forward networks and recurrent networks with their gated variants (LSTM and GRU) are evaluated and it was confirmed that these neural networks do not learn abstract grammar rules as expected.
- RBP (Relation Based Patterns) method as an inductive bias has been proposed to enable the learning of abstract grammatical patterns within the neural network structures.

- RBP can be integrated into standard neural network architectures in early, mid and late fusion settings.
- the networks with suitable RBP structure learned the abstract grammar patterns with 100% accuracy.
- the integration of RBP to neural network models improved the performance in neural language modelling tasks along with artificial grammar learning tasks which proves that RBP can be expanded to other sequential rule learning tasks as well.

RBP is one such method of adding inductive bias to the standard neural networks. There are other experiments with variants of RBP and different hyperparameter configurations to evaluate and understand the effect of RBP as an inductive bias within the standard neural network models which is beyond the scope of this article. For more detailed description of the RBP approach and experiment details including other extended works, please refer our papers.[14,25,24]

In future this work can be extended towards improving the performance of neural networks in providing better abstractions and generalizations for other forms of abstract relations on more complex tasks such as question answering or perception-based reasoning and other relational learning tasks.

References

[1] Raquel G. Alhama and Willem Zuidema. "Pre-Wiring and Pre-Training: What does a neural network need to learn truly general identity rules". In: *CoCo at NIPS* (2016).

[2] Gerry Altmann. "Learning and development in neural networks the importance of prior experience". In: *In Cognition* 85(2) (2002), B43–B50.

[3] Gerry Altmann and Zoltan Dienes. "Technical comment on rule learning by seven-month-old infants and neural networks". In: *In Science* 284(5416) (1999), pp. 875–875.

[4] David G. T. Barrett et al. *Measuring abstract reasoning in neural networks*. 2018. arXiv: `1807.04225 [cs.LG]`.

[5] Peter W Battaglia et al. "Relational inductive biases, deep learning, and graph networks". In: *arXiv preprint arXiv:1806.01261* (2018).

[6] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[7] Rodney A. Brooks. "Intelligence without representation". In: *Artificial Intelligence : https://doi.org/10.1016/0004-3702(91)90053-M* 47 (1991), pp. 139–159.

[8] Morten H Christiansen and Suzanne L Curtin. "The power of statistical learning: No need for algebraic rules". In: 114 (1999), p. 119.

[9] Z. Dienes and J. Perner. "A theory of implicit and explicit knowledge." In: *Behavioural and Brain Sciences, 22, 735–755.* (1999).

[10] P. Dominey and F. Ramus. "Neural network processing of natural language: ISensitivity to serial, temporal and abstract structure of language in the infant." In: *Language and Cognitive Processes* (2000), 15(1), 87–127.

[11] Jeffrey Elman. "Generalization, rules, and neural networks: A simulation of Marcus et. al". In: (1999). URL: `%7Bhttps://crl.ucsd.edu/%5C~%7B%7Delman/Papers/MVRVsimulation.html%7D`.

[12] Reuben Feinman and Brenden M. Lake. *Learning Inductive Biases with Simple Neural Networks*. 2018. arXiv: `1802.02745 [cs.CL]`.

[13] A. Kinder. "The knowledge acquired during artificial grammar learning: Testing the predictions of two connectionist models." In: *Psychological Research, 63, 95–105.* (2000).

[14] Radha Kopparti and Tillman Weyde. "Factors for the Generalisation of Identity Relations by Neural Networks". In: *ICML Workshop on Understanding and Improving Generalization in Deep Learning.* (2019). URL: `https://arxiv.org/abs/1906.05449`.

[15] Brenden Lake and Marco Baroni. “Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks”. In: *International Conference on Machine Learning*. 2018, pp. 2879–2888.

[16] Brenden Lake et al. “Building Machines That Learn and Think Like People”. In: *CoRR* abs/1604.00289 (2016). URL: `http://arxiv.org/abs/1604.00289`.

[17] G. F. Marcus. “Deep Learning : a critical appraisal”. In: *arXiv:1801.00631* (2018).

[18] G. F. Marcus. “The Algebraic Mind: Integrating Connectionism and cognitive science.” In: *Cambridge MIT Press* (2001).

[19] G. F. Marcus et al. “Rule learning by seven-month-old infants.” In: *Science, 283* 5398 (1999), pp. 77–80.

[20] Jeff Mitchell et al. “Extrapolation in NLP”. In: *arXiv:1805.06648* (2018).

[21] H. Schaffrath. “The Essen Folksong Collection in the Humdrum Kern Format”. In: *Database edited by David Huron, CCAHR, Menlo Park, CA.* (1995). Ed. by D. Huron. URL: `http://kern.ccarh.org/cgi-bin/ksbrow`.

[22] Mark Seidenberg and Jeffrey Elman. “Do infants learn grammar with algebra or statistics?” In: *Science* 284.5413 (1999), pp. 433–433.

[23] Shastri and Chang. “A Spatiotemporal Connectionist Model of Algebraic Rule-Learning”. In: *International Computer Science Institute* (1999), TR-99-011.

[24] Tillman Weyde and Radha Kopparti. “Feed-Forward Neural Networks need Inductive Bias to Learn Equality Relations.” In: *Relational Representation Learning Workshop, NeurIPS* (2018).

[25] Tillman Weyde and Radha Kopparti. “Modelling Identity Rules with Neural Networks”. In: *https://arxiv.org/abs/1812.02616* (2018).