Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)

Visualisation and 'Diagnostic Classifiers' Reveal how Recurrent and Recursive Neural Networks Process Hierarchical Structure* (Extended Abstract)

Dieuwke Hupkes and **Willem Zuidema**
ILLC, University of Amsterdam
d.hupkes@uva.nl, zuidema@uva.nl

Abstract

In this paper, we investigate how recurrent neural networks can learn and process languages with hierarchical, compositional semantics. To this end, we define the artificial task of processing nested arithmetic expressions, and study whether different types of neural networks can learn to compute their meaning. We find that simple recurrent networks cannot find a generalising solution to this task, but gated recurrent neural networks perform surprisingly well: networks learn to predict the outcome of the arithmetic expressions with high accuracy, although performance deteriorates somewhat with increasing length. We test multiple hypotheses on the information that is encoded and processed by the networks using a method called *diagnostic classification*. In this method, simple neural classifiers are used to test sequences of predictions about features of the hidden state representations at each time step. Our results indicate that the networks follow a strategy similar to our hypothesised 'cumulative strategy', which explains the high accuracy of the network on novel expressions, the generalisation to longer expressions than seen in training, and the mild deterioration with increasing length. This, in turn, shows that diagnostic classifiers can be a useful technique for opening up the black box of neural networks.

1 Introduction

A key property of natural language is its hierarchical compositional semantics: the meaning of larger wholes depends not only on its words and subphrases, but also on the way they are combined. Subphrases, in turn, can also be composed of smaller subphrases, resulting in sometimes quite complex hierarchical structures. For example, consider the meaning of the noun phrase *the student who looked at machine learning case studies*, which is a combination of the meanings of *the student*, *looked at* and *machine learning case studies*. The meaning of the latter phrase – a compound noun – is a combination of the meanings of *machine learning* and *case studies*, which are combinations of the meanings of the individual words. Such hierarchical structures can be well represented by symbolic models [Chomsky, 1956; Montague, 1970], but if and how they can be represented by artificial *neural* models is an open question. This question has a long tradition in linguistics and computational neuroscience; recently, it has also received much attention from the field of natural language processing (NLP).

As we argue in the full paper, neither theoretical results [Siegelmann and Sontag, 1995], nor handcrafted examples [Gers and Schmidhuber, 2001; Rodriguez, 2001], nor large-scale NLP applications [Sutskever *et al.*, 2014; Bahdanau *et al.*, 2015; Kalchbrenner *et al.*, 2014; Socher *et al.*, 2013; Le and Zuidema, 2015; Roth and Lapata, 2016] provide a convincing demonstration that artificial neural networks are adequately computing the meaning of sentences with hierarchical structure or help us understand how hierarchy and compositionality can be implemented in large collections of interconnected artificial neurons.

A deeper insight into the internal dynamics of artificial neural networks while they process hierarchical structures could prove valuable both from a theoretical and a cognitive perspective, and is – considering the difficulty of searching through the vast parameter space of high dimensional neural networks – also interesting from an engineering point of view. However, such insight is not easily acquired by studying natural language with all its complexities directly. In this paper, we take a different approach. To analyse the mechanism of processing hierarchical compositional structures isolated from all other aspects involved in processing language, we study an artificial language – the language of arithmetics – of which both sentences, phrases and lexical items have clearly defined meanings. We investigate if and how simple recurrent networks (SRNs) [Elman, 1990] and gated recurrent

*This paper is an extended abstract of a paper published in the Journal of Artificial Intelligence Research [Hupkes *et al.*, 2018]. In addition to recurrent neural networks, in this paper also recursive neural networks [Socher *et al.*, 2010] are discussed.

units (GRUs) [Cho *et al.*, 2014] can learn to correctly compute the meaning of sentences from this language (Section 2). In Section 3, we present a thorough analysis of how they process them. For this analysis, we use a technique called *diagnostic classification* [Hupkes *et al.*, 2018]. We conclude in Section 4.

2 Methods

In this section, we define the artificial language we are studying, followed by a description of our experiments.

2.1 Arithmetic Language

The vocabulary of the artificial language we consider consists of words for all integers in the range $\{-10, \ldots, 10\}$, the operators `plus` and `-` and the brackets `(` and `)`. The grammatically correct phrases – i.e. sequences of words – in this *arithmetic language* comprise all grammatically correct, fully bracketed arithmetic expressions that can be formed with these symbols. The meaning of an expression is the solution of the arithmetic expression that it represents. For instance, the meaning of the phrase `( ten minus ( five plus three ))` is 2.[1]

We refer to expressions and sets of expressions by using the number of numeral words they contain. For instance, **L5** refers to all expressions with exactly 5 numerals and **l5** is an expression belonging to **L5**. Some examples can be found in Table 1.

L1	`one`, `-three`, `nine`
L2	`( five plus three )`
L3	`( ( seven minus -eight ) minus six )`
L4	`( ( ( -two minus six ) plus one ) plus ten)`

Table 1: Sentences from different subsets of the arithmetic language. Both numerals, operators and brackets are treated as words; words are represented by n-dimensional numerical vectors.

The arithmetic language is specifically chosen to allow us to study the mechanism of hierarchical compositionality in isolation, separate from other important aspects of natural language, such as structural and lexical ambiguity, irregular paradigms, multi-word units and idiomatic expressions. Furthermore, the symbolic nature of the arithmetic language allows us to formulate precise strategies to compute the meaning of expressions, which can be used to aid analysis of the dynamics of the internal dynamics of a network processing sentences, as we will see later.

2.2 Training and Performance

We train 40 models with 15 hidden units and an embedding size of 2 to predict the outcome of a randomly sampled subset of expressions from **L1**, **L2**, **L4**, **L5** and **L7** (3000 from each set). We test all models on a large sample of unseen expressions from these subsets, as well as expressions from lengths that were not seen at all during training (**L3**, **L6**, **L8** and **L9**). This way, we test generalisation to both unseen sequences and unseen syntactic structures.

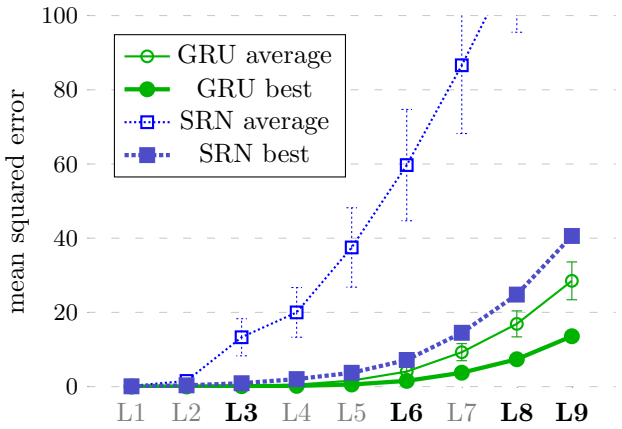


Figure 1: Average and best mean squared error for 20 GRU and 17 SRN models. Three other trained SRN models that did not learn to capture any structural knowledge were excluded from the plot. Error bars indicate standard error. Sentences lengths that were not included in the training set are bold-faced.

Of the 20 trained SRN models, three did not learn to capture any structural knowledge, reflected by a high error for short (but unseen) sentences with three numerals (**L3**). It is unclear to what extent the remaining 17 SRN models learned solutions incorporating the syntactic structure of sentences. Most GRU models, on the other hand, show a convincing ability to generalise, with a mean squared error that slowly increases with the length of the sentence. A summary of the results is plotted in Figure 1.

3 Analysis

Despite their frequent use, the internal dynamics of recurrent networks largely remain a black box, which limits their usefullnes as explanatory models of the underlying task, but which also hinders progress in developing more sophisticated training methods. Most approaches to gain more insight in the workings of recurrent networks are based on visual inspection. For example, Karpathy *et al.* (2015) and Li *et al.* (2016) study cell and gate activations under different conditions; Tang *et al.* (2017) plot distributions of cell activations and temporal traces of t-SNE cell vectors of a speech recognition RNN; and Strobelt *et al.* (2016) present a tool[2] that facilitates visual analysis of gated RNNs. Although such methods give intriguing clues, the potential conclusions that can be

[1] Throughout this paper, we will often abbreviate the full forms such as `left_bracket five plus three right_bracket` as `( 5 + 3 )`.

[2] LSTMVis, http://lstm.seas.harvard.edu/, is a visualisation tool that can be applied to gated RNNs, including the GRU model.

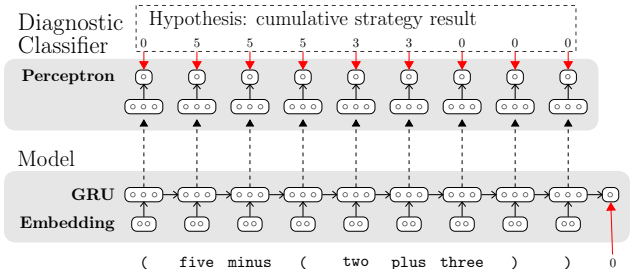


Figure 2: Testing a single hypothesis with a diagnostic classifier.

drawn from visual inspection are qualitative rather than quantitative, and they concern only small parts of the network's overall behaviour. Additionally, visualisation-based methods are typically restricted to finding functions or features that are encoded by one cell, while being insensitive to operations distributed over multiple cells or cells that encode multiple features at the same time.

3.1 Diagnostic Classification

In this paper, we use an alternative approach called *diagnostic classification* [Hupkes *et al.*, 2018]. This approach is based on the idea that if a model is computing or keeping track of certain information it should be possible to extract this information from its internal state space. Whether a network is representing a certain variable or feature is then tested by training an additional classifier – a *diagnostic classifier* – to predict the sequence of values this variable takes at each step of the computation from the sequence of hidden states a trained network goes through while processing the input sentence. If the sequence of values can be predicted with a high accuracy by the diagnostic classifier, this indicates that the hypothesised information is indeed computed by the network. Conversely, a low accuracy suggests that this information is not represented in the hidden state.

Diagnostic classification is a generic method that addresses most of the shortcomings we listed for visualisation-based methods. The approach, which bears similarities with analysis methods presented by Adi *et al.* (2017), Gelderloos and Chrupała (2016) and Alain and Bengio (2017), can be used to quantitatively test hypotheses about neural networks that range from very simple to fully fledged strategy descriptions. For instance, it could be used to test for the existence of feature detectors such as the inside-quote detectors found by [Karpathy *et al.*, 2015], but it can also be extended to test whether a network is computing the type of information needed for an algorithmically defined symbolic strategy.

3.2 Symbolic Strategies

We use diagnostic classifiers to probe the strategy the trained networks are implementing on an algorithmic level [Marr, 1982]. Perhaps the most obvious candidate for a such a strategy involves traversing through the expression, computing the outcome of all subtrees until an outcome for the full tree is reached. To do this in an incremental fashion, the intermediate result of the computation of the current subtree should be pushed onto a stack whenever a new, smaller subtree begins. At that point, also the operator that will later be used to integrate the outcome of the newly started subtree with its parent, should be stored on a stack. We refer to this strategy with the name *recursive strategy*.

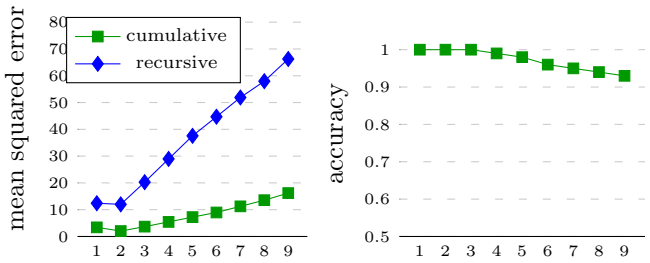
Alternatively, the digits can be accumulated immediately at the moment they are encountered. This means that at any point during the computation a prediction of the solution of the expression is maintained. Consequently, this strategy – which we call the *cumulative strategy* – does not require a stack with previous results, but it does require keeping track of previously seen operators to decide whether the next number should be added or subtracted when a bracket closes.

These two strategies result in very different predictions about the intermediate results stored (and computed) during processing a sequence. For instance, after seeing the word `three` in the sequence `( five minus ( ( two minus three ) plus seven ) )`, the recursive strategy should have a representation of the value within the current brackets (which is -1), whereas the cumulative strategy should maintain a representation of the value of the expression up to that point (6). Furthermore, the cumulative strategy requires knowledge of whether the next encountered digit should be added or subtracted (a variable we refer to with the name `mode`), which is a variable that plays no role in the recursive strategy. A sketch of the diagnostic classification method applied to test if a network computes the intermediate values of the cumulative strategy is depicted in Figure 2.

3.3 Results

To test whether our trained networks are following either the cumulative or recursive strategy, we train diagnostic classifiers to predict the sequences of intermediate results of both these strategies, as well as the `mode` used by the cumulative strategy to determine whether the next encountered digit should be added or subtracted. As the diagnostic model should merely read out whether certain information is present in the hidden representations rather than perform complex computations itself, we use a simple linear model as diagnostic classifier.

Strategy results We find that the values required for the cumulative strategy can be more accurately predicted than the intermediate recursive strategy values (see Figure 3). From these findings it appears unlikely that the network implements a fully recursive strategy employing a stack of intermediate results. For the cumulative strategy the predictions are generally accurate, even for longer sentences. The same is true for the `mode` variable of the cumulative strategy, which can be predicted almost perfectly for sentences up until length 5 (with accuracies in the range of 0.98 – 1.0), but is also



(a) `result` prediction (b) cumulative `mode` prediction

Figure 3: Results of diagnostic models for a GRU model on different subsets of languages.

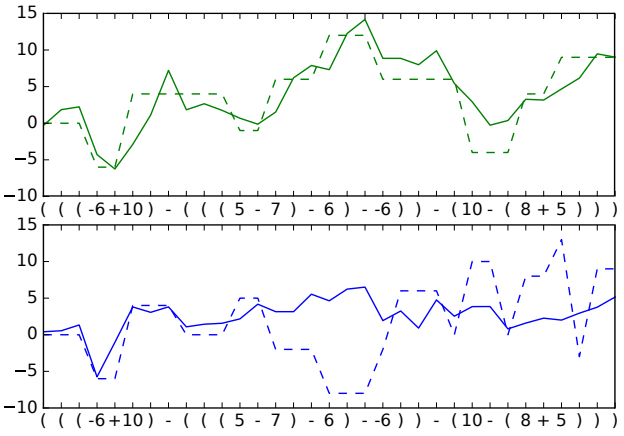


Figure 4: Trajectories of the cumulative (green, upper) and recursive (blue, lower) classifier, along with their targets trajectories for the `result` values. Dashed lines show target trajectories.

accurately kept track of for longer sequences (an accuracy of 0.93 for **L9** sentences).

Plotting trajectories Aside from evaluating the overall match with a specific hypothesis, we can also track the fit of predictions over time, by comparing the trajectories of predicted variables with the trajectories of observed variables while the networks process different sentences. In Figure 4, the predictions of the diagnostic classifiers on a randomly picked **L9** sentence is depicted, along with the target trajectories as defined by the hypotheses. These trajectories confirm that the curve representing the cumulative strategy is much better predicted than the recursive one. A correlation test over 5000 **L9** sentences shows the same trend: Pearson's $r = 0.52$ and 0.95 for recursive and cumulative, respectively. For a more elaborate analysis of of how diagnostic classifiers can be used to increase insight in the internal dynamics of recurrent neural networks, including gate values, we refer to the work presented by Hupkes and Zuidema (2017).

We also observe an important qualitative difference between the diagnostic classifier trajectories and the target values: The diagnostic classifier trajectories are smooth, changing value at every point in time, whereas the target trajectories are jumpy and often stay on the same value for longer time spans. This indicates that a refinement of the symbolic cumulative hypothesis, in which information is integrated more gradually, would be more suitable for a network like this.

4 Conclusion

In this paper we studied how recurrent neural networks process hierarchical structures, using an arithmetic language as a convenient, idealised task with unambiguous syntax and semantics and a limited vocabulary. As it turns out, gated recurrent networks can learn to compute the meaning of arithmetic expressions and generalise to longer expressions than seen in training, whereas simple recurrent networks cannot.

Using diagnostic classifiers, we were able to analyse the internal dynamics of the GRU network in more detail than ever before. This allows us to conclude that the GRU is achieving its suprisingly good accuracy and generalisation behaviour by following a strategy that roughly approximates the symbolic 'cumulative strategy'. From this we learn something about how neural networks may process languages with a hierarchical compositional semantics and, perhaps more importantly, also provide an example of how we can *open the black box* of the many successful deep learning models in natural language processing (and other domains) when visualisation alone is not sufficient.

References

[Adi *et al.*, 2017] Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

[Alain and Bengio, 2017] Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *Proceedings of the 5th International Conference on Learnign Representations (ICLR) – Workshop Track*, 2017.

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, 2014.

[Chomsky, 1956] Noam Chomsky. Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124, 1956.

[Elman, 1990] Jeffrey L Elman. Finding Structure in Time. *Cognitive Science*, 14(2):179–211, 1990.

[Gelderloos and Chrupała, 2016] Lieke Gelderloos and Grzegorz Chrupała. From phonemes to images: levels of representation in a recurrent neural model of visually-grounded language learning. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1309–1319, 2016.

[Gers and Schmidhuber, 2001] Felix A Gers and Jürgen Schmidhuber. LSTM recurrent networks learn simple context-free and context-sensitive languages. *Neural Networks, IEEE Transactions on*, 12(6):1333–1340, 2001.

[Hupkes and Zuidema, 2017] Dieuwke Hupkes and Willem Zuidema. Diagnostic classification and symbolic guidance to understand and improve recurrent neural networks. In *Proceedings of Neural Information Processing Systems – Workshop track*, 2017.

[Hupkes *et al.*, 2018] Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research (JAIR)*, 61, 2018.

[Kalchbrenner *et al.*, 2014] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 655—-665, 2014.

[Karpathy *et al.*, 2015] Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks. In *Proceedings of the International Conference on Learning Representations 2016*, pages 1–13, 2015.

[Le and Zuidema, 2015] Phong Le and Willem Zuidema. The forest convolutional network: compositional distributional semantics with a neural chart and without binarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1155–1164, 2015.

[Li *et al.*, 2016] Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. Visualizing and understanding neural models in NLP. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 681–691, 2016.

[Marr, 1982] David Marr. Vision: A computational investigation into the human representation and processing of visual information. *Phenomenology and the Cognitive Sciences*, 8(4):397, 1982.

[Montague, 1970] Richard Montague. Universal grammar. *Theoria*, 36(3):373–398, 1970.

[Rodriguez, 2001] Paul Rodriguez. Simple recurrent networks learn context-free and context-sensitive languages by counting. *Neural computation*, 13(9):2093–118, 2001.

[Roth and Lapata, 2016] Michael Roth and Mirella Lapata. Neural semantic role labeling with dependency path embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1192–1202. Association for Computational Linguistics, 2016.

[Siegelmann and Sontag, 1995] Hava T. Siegelmann and Eduardo D. Sontag. On the computational power of neural nets. *Journal of Computer and System Sciences*, 50(1):132–150, 1995.

[Socher *et al.*, 2010] Richard Socher, Christopher D Manning, and Andrew Y Ng. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *NIPS 2010: Deep Learning and Unsupervised Feature Learning Workshop*, pages 1–9, 2010.

[Socher *et al.*, 2013] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, Christopher Potts, et al. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer, 2013.

[Strobelt *et al.*, 2018] Hendrik Strobelt, Sebastian Gehrmann, Bernd Huber, Hanspeter Pfister, and Alexander M Rush. Visual analysis of hidden state dynamics in recurrent neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):667–676, 2018.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.

[Tang *et al.*, 2017] Zhiyuan Tang, Ying Shi, Dong Wang, Yang Feng, and Shiyue Zhang. Memory visualization for gated recurrent neural networks in speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 2736–2740. IEEE, 2017.