Exploring the Limits of Encoder-Decoder Networks with regard to Sorting Tasks

By: Robert Walters
Professor: Dr. Seo
Advisor: Dr. Phillips

Abstract

Machine learning is becoming increasingly popular among researchers of all stripes, due to the usefulness of trained artificial neural networks for accomplishing many disparate tasks. A particular network called the encoder-decoder network is most useful for translating variable-length sequences to fixed-length vectors, but how does it do when given sorting tasks? The goal of our project is to answer this question, as well as the more complex variations of it. Can it sort variable-length lists of words into alphabetical order? Can it sort lists of words with numbers included in the mix? What about sorting numbers and words at the same time? The primary purpose is to study the accuracy and speed with which they accomplish the tasks, and what variations make it easier and more difficult. We train encoder-decoder networks to sort numbers - first horizontally and then vertically, and finally vertically for only the first column. The latter test is a proof-of-concept for sorting the first letter of an ASCII list of words. We anticipate finding that the encoder-decoder network to be best at the first task and worst at the last task, but still accomplish it with a reasonable accuracy level. Future work will involve attempting to design an artificial computational model of working memory, and using the best version of the alphabetizing encoder-decoder as a framework for teaching this larger system the same task.

Introduction

Artificial Neural Networks (ANNs) have taken the field of computer science by storm, promising a bright future in AI and automation. A specific type of recurrent neural network (RNN) called an RNN Encoder-Decoder system - which we generally refer to just as an Encoder-Decoder network for the purposes of this paper - was created for the purposes of machine translation: taking one language (i.e. English) and translating it to another (i.e. French). Investigating how Encoder-Decoder networks handle sorting tasks, such as the Letter and Number Sequencing task, is an important piece for a future addition to an existing software designed to model working memory in humans

The aims for the project are:
1) Explore the limitations of Encoder-Decoder networks
2) Train an Encoder-Decoder network to tackle the Letter and Number Sequencing Task

References

Modeling Situations in Neural Chat Bots, 2017. doi: doi.org/10.18653/v1/P17-3020.

Kyunghyun Cho, Bart van Merrienboer, C ̧aglar G ̈ul ̧cehre, Fethi Bougares, Hol-ger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR, abs/1406.1078,2014. URLhttp://arxiv.org/abs/1406.1078.

Simon F. Crowe. Does the letter number sequencing task measure anything more than digit span?Assessment, 2000. doi: 10.1177/107319110000700202.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MITPress, 2016.http://www.deeplearningbook.org.

R ́emi Lebret, Pedro H. O. Pinheiro, and Ronan Collobert. Phrase-based image captioning.CoRR, abs/1502.03671,2015.URLhttp://arxiv.org/abs/1502.03671.

Marta K. Meilicki, Rebecca H. Koppel, Gabriela Valencia, and Jennifer Wiley. Measuring working memory capacity with the letter-number sequencing task:Advantages of visual administration. Applied Cognitive Psychology, 2018.4

George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Review, 1956. doi:10.1037/h0043158.

Micheal A. Nielsen.Neural Networks and Deep Learning. Determination Press,2015.

Randall C. O'Reilly, Yuko Munakata, Michael J. Frank, Thomas E. Hazy, and Contributors. Computational Cognitive Neuroscience. Wiki Book, 1st Edition,2012. URLhttp://ccnbook.colorado.edu.

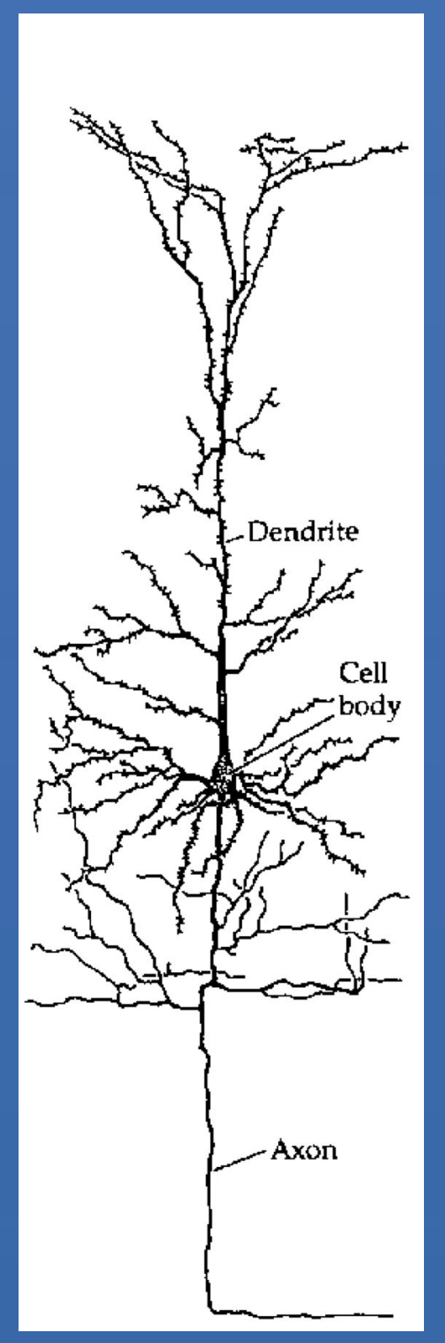
Joshua L. Phillips and David C. Noelle. A biologically inspired working memory framework for robots. IEEE International Workshop on Robots and Human Interactive Communications, 2005.

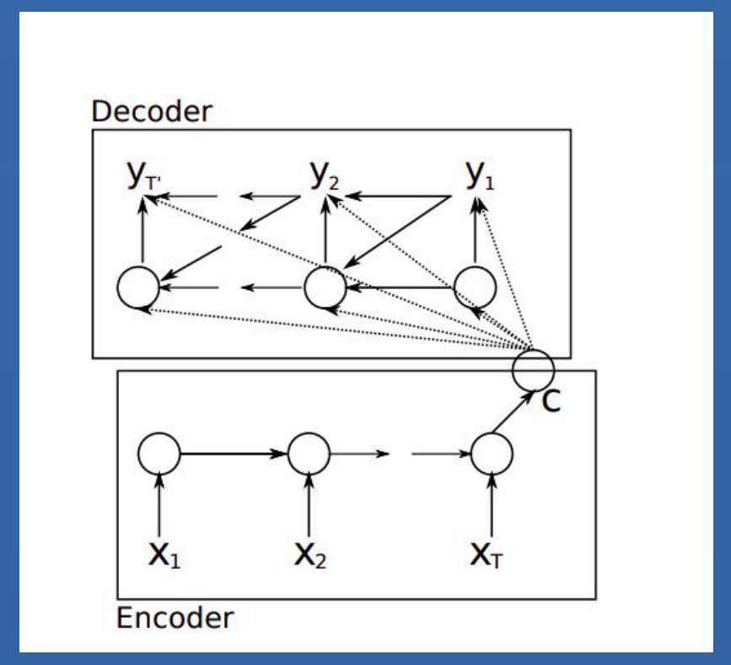
Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. InAdvances in neural information processing systems,pages 3104–3112, 2014.

Nicholas S. Thaler, Gerald Goldstein, Jay W. Pettegrew, James F. Luther,Cecil R. Reynolds, and Daniel N. Allen. Developmental Aspects of Working and Associative Memory. Archives of Clinical Neuropsychology, 28(4):348–355, 01 2013.ISSN 0887-6177.doi: 10.1093/arclin/acs114.URLhttps://doi.org/10.1093/arclin/acs114.

Background

The most basic component of an Artificial Neural Network, the neuron, was designed to be a computational analogue to the real biological neurons we have in our nervous systems. Like real neurons, there are a series of inputs on one with, with one output (which itself may be wired to many downstream neurons). In a real neuron, each input has a signal strength (to oversimplify the matter) which together cause a charge to build up in the axon until a threshold is reached and the neuron "fires", or sends its own signal to the following neurons via its dendritic spines. Similarly, in a computational neuron the various inputs are multiplied by their own weight, an analogue to signal strength, and then summed together and fed into some activation function which determines behavior around a threshold, similar to the neuron bias value. The function of a single neuron could be described as a detector, although what it detects is extremely context-dependent and even malleable. Both in the brain and in our software these neurons are organized into layers, which themselves connect to other layers to form complex neural networks and/or brain regions with localized function specialization. Over time, these neural networks can "learn" by essentially finding the (local) minima or maxima of some complicated multi-dimensional function describing the abstract concept they are attempting to model.





The type of neural network system we focus on is called the Encoder-Decoder. This system, also called sequence-to-sequence or seq2seq network, is built using Long Short-Term Memory (LSTM) layers organized in a particular way. LSTM neurons are complex neurons which often serve as a means of mapping to and from a fixed dimensionality. The nomenclature describes its primary purpose: Encoding a variable-length sequence into a fixed-length vector and doing the opposite with a decoder. These networks have been shown to perform well on tasks just as machine language translation, image captioning, conversational modeling, and many others.

There are many tests used in the medical field to gauge working memory ability in humans, but one of the most thorough at testing working memory *specifically* is the Letter and Number Sequencing (LNS) task, which is part of the WAIS III assessment. Loss or reduction in working memory ability can suggest disease, mental decline, or brain damage. When given to people, the participants must listen to a random list of numbers and letters, and then put the numbers in ascending order and the letters in alphabetical order. This may seem like a simple task to us, but it involves a complex interplay between brain regions; especially treating letters and numbers differently during the same task. Depending on how long the list is, even we might have to pause and focus!

Methods

- As with any large task, it is useful to break it down into chunks. Our first chunk is to build an encoder-decoder network and train it to complete a simple row-wise ascending sort. All programming for this project was done in Python 3.

```
[[ 7 26  8 23 16 16 21 28  7 18]
 [23 26 19  3 12 17 26  1 29 22]]
```

```
[[ 7  7  8 16 16 18 21 23 26 28]
 [ 1  3 12 17 19 22 23 26 26 29]]
```

- Our second, more difficult, task is to simulate the Letter and Number Sequencing task from the WAIS III. We do this by letting 1-9 represent "numbers" and 10-36 represent "letters" in our array. To differentiate the two, we sort the "numbers" in *descending* order, and the "letters" in ascending order – in the same array.

```
[[29 21  2 34 31  7 35  2  4 22]
 [27 28  6 18  3  7 18  4 30 22]]
```

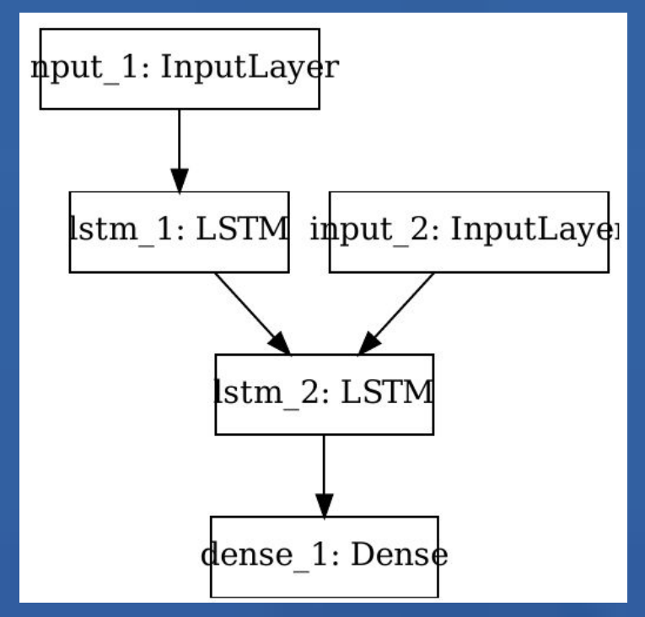
```
[[ 7  4  2  2 21 22 29 31 34 35]
 [ 7  6  4  3 18 18 22 27 28 30]]
```

- Our final and most difficult task is to train a network for word alphabetization. We attempt to accomplish this by pretending each row is a word – perhaps of ascii-encoded values – that we need to alphabetize, out of a list of several (in this case, 10). We then sort the arrays in ascending order by the first column (keeping the row-wise order intact).

```
[[35 22 25 16 31  6 27  9 28  2]
 [12  4 19 27 21 24 13  3 27 22]
 [17  3 29 20 12 35 23  6 25 32]
 [18  7 13  6 33 10 22  3 12 26]
 [17 19 22 32 33  3 10 34  4 35]
 [35 34 19 32 25 31  4 10  5 22]
 [ 1 14 34 15 16 19  1 23 35 24]
 [28  7 10 32 11  8 32  8 35 25]
 [ 4 10 30 33 14 35 14 28 18  3]
 [26 13 11 15 28 23 15 28 12 35]]
```

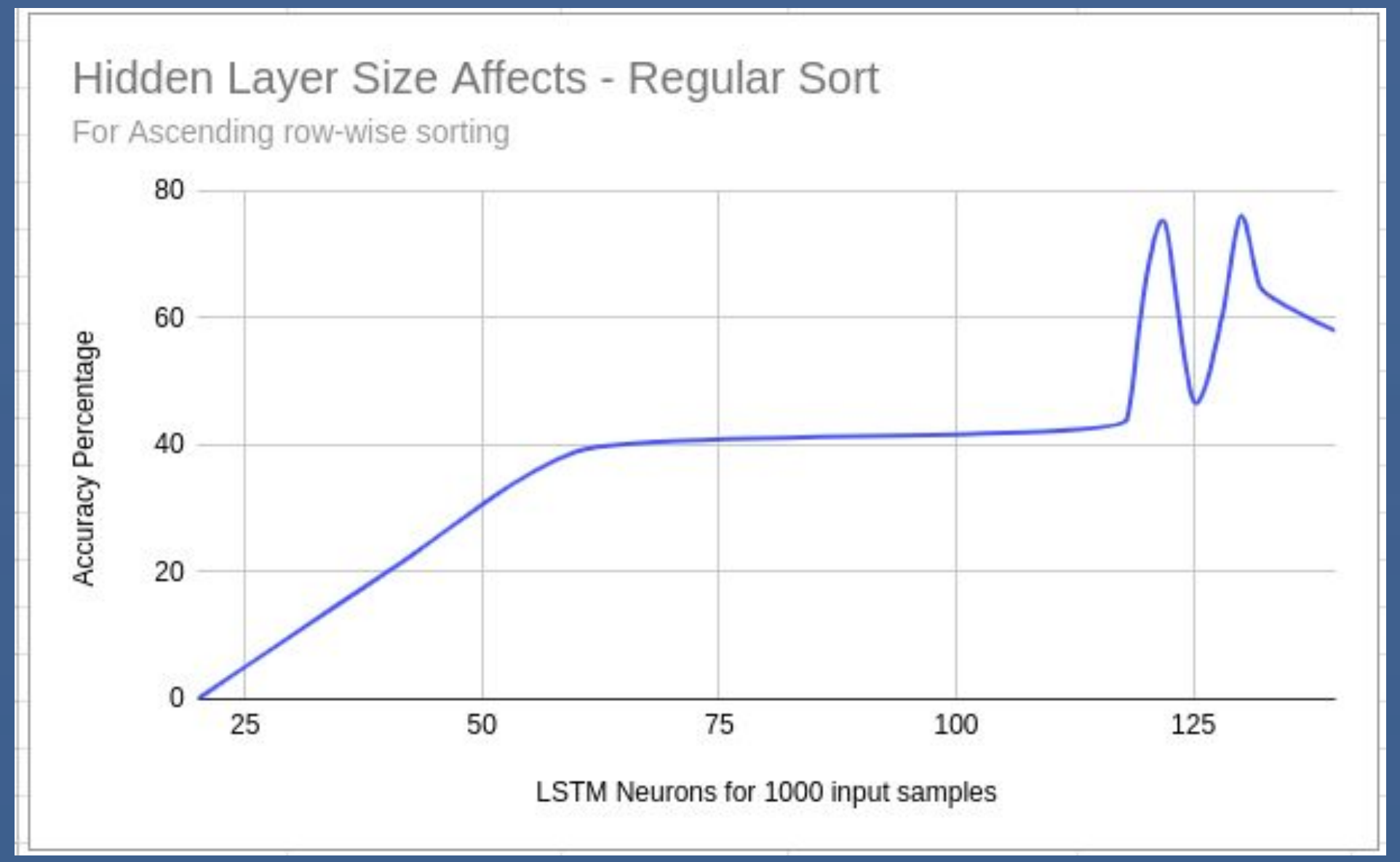
```
[[ 1 14 34 15 16 19  1 23 35 24]
 [ 4 10 30 33 14 35 14 28 18  3]
 [12  4 19 27 21 24 13  3 27 22]
 [17 19 22 32 33  3 10 34  4 35]
 [17  3 29 20 12 35 23  6 25 32]
 [18  7 13  6 33 10 22  3 12 26]
 [26 13 11 15 28 23 15 28 12 35]
 [28  7 10 32 11  8 32  8 35 25]
 [35 34 19 32 25 31  4 10  5 22]
 [35 22 25 16 31  6 27  9 28  2]]
```

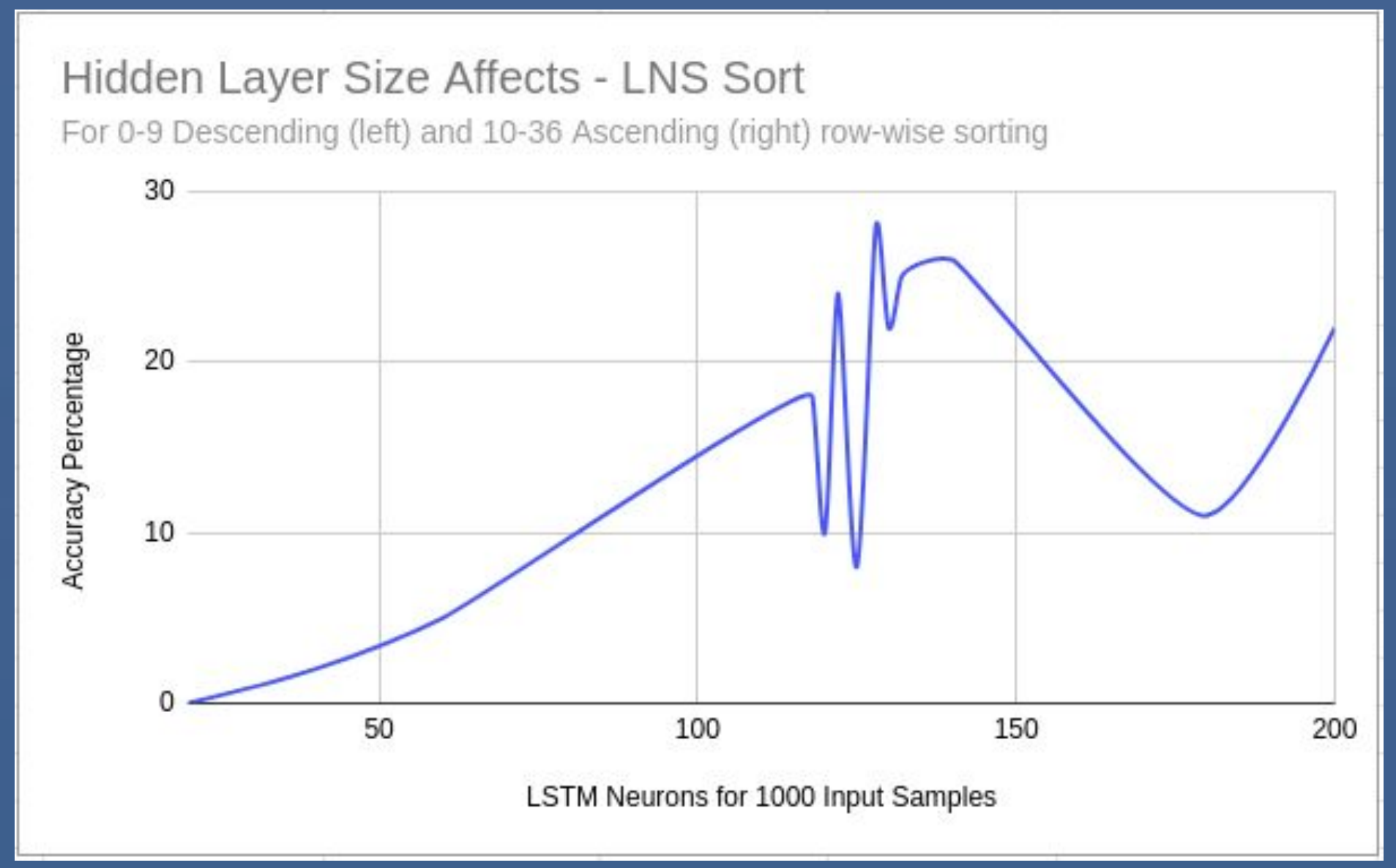
- The network itself has three layers, excluding the two input layers. Built using keras, we put two LSTM layers and a final normal "dense" layer. The two LSTM layers are the true encoder and decoder networks – the star of our study.



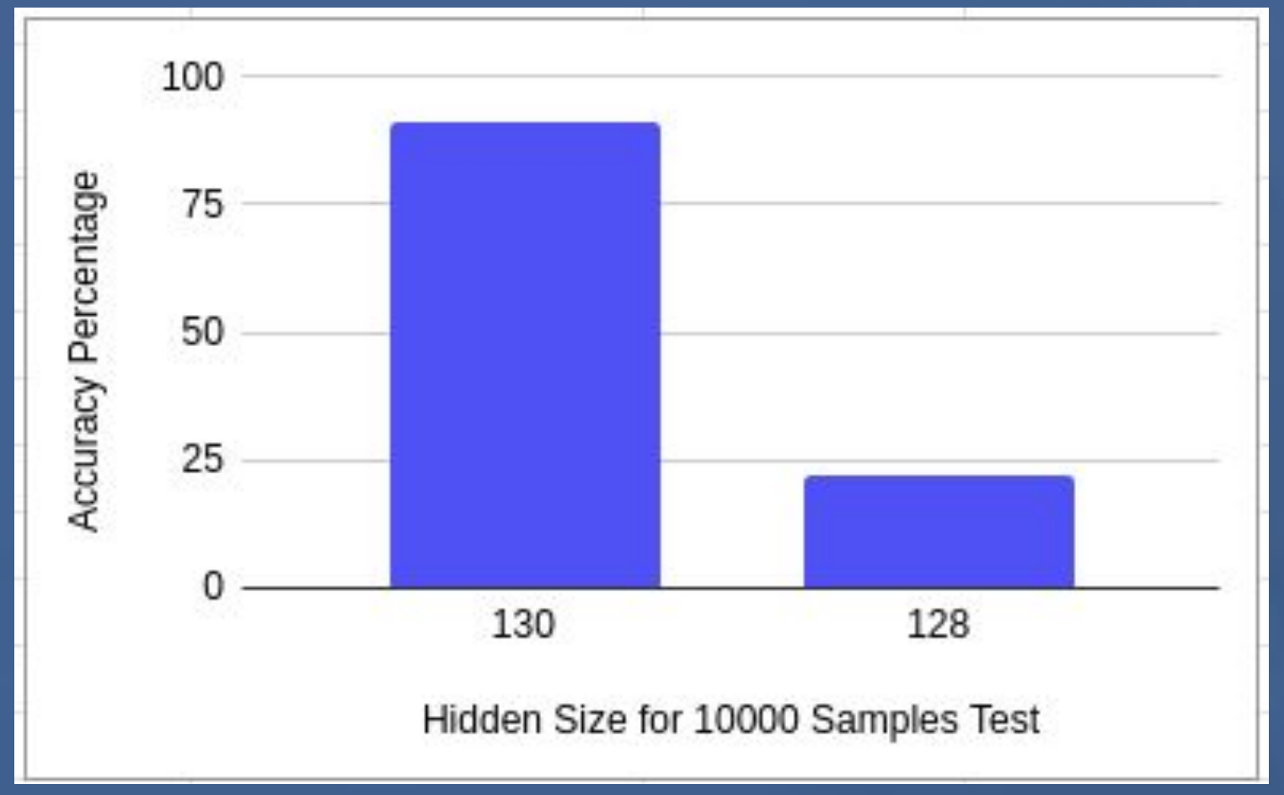
Results

- One of most important features for improving accuracy was
- dialing in the best amount of LSTM units in the hidden layers.





- The other major factor was sample size, although the
- improvements to the LNS sort were slim to none.



- We were unable to generate meaningful results from the the alphabetization task within the scope of this project, and the network has yet to surpass 0% accuracy outside of teacher-forcing.