# Reproducibility Study of “RNNs of RNNs: Recursive Construction of Stable Assemblies of Recurrent Neural Networks”

**Laura De Los Santos**
AI Model Share, Columbia University
New York, NY
laura.santos@columbia.edu

## Reproducibility Summary

### Scope of Reproducibility

In this work, we study the reproducibility of the paper RNNs of RNNs: Recursive Construction of Stable Assemblies of Recurrent Neural Networks by Kozachkov, Ennis, and Slotine to verify their main claims of a proof to preserve stability in neural network models with subnetworks. Further, the authors claim to have achieved new state of the art results while imposing the stability constraint and increasing sparsity in the RNNs of RNNS structure.

### Methodology

The authors of the paper provide the implementation of RNNs of RNNs training in PyTorch and made the code available on GitHub. We modified the code accordingly to change the dataset used for training and test all 3: sequential MNIST, permuted sequential MNIST, and sequential CIFAR. Otherwise, the same hyperparameters and architecture as mentioned for trial #10 were used. The experiments were run on NVIDIA Quadro P5000 GPUs. Our reproducibility study comes at a total computational cost of 96 GPU hours.

### Results

We validated the claims of the paper about maintaining stability on the “network of networks” based on the proposed novel constraints. We were able to reproduce the test accuracy results with less than a 1.5% deviation for the CIFAR dataset, where for the MNIST and permuted MNIST, the deviation was less than a 0.25%. Note that this replication study did not include all 10 experiments for each set due to computational limitations.

### What was easy

The documentation for the original paper provides well commented code and publicly available datasets used to train and validate the model. An extensive appendix explains the theoretical aspects of the paper for the stability proofs and provides sufficient background for the reader to understand.

### What was difficult

The experiments required access to powerful computing resources and an up-front cost to purchase GPU access. Small code changes were required to ensure that the correct dataset was being evaluated and it also required troubleshooting different versions of Python and PyTorch to avoid CUDA-related issues.

### Communication with original authors

We emailed the authors regarding their initial hypothesis about how this network architecture would perform on the

CIFAR dataset, and also proposed alternatives to Google Colab limitations that we also had to overcome during the replication process.

**RESULTS**

In order to reproduce the results, we used Paperspace + Gradient Notebooks as it is more economical than Google Colab and it does not have a 24 hour time limit either. If needed, notebooks can run up to 7 days.

For the Seq MNIST, PerSeqMNIST, and SeqCIFAR the running time was 28 hours, 24 hours, 36 hours, respectively. These are slightly higher than the time claimed in the paper, which could be due to different GPUs being used and extra time spent aligning the required Python and PyTorch versions needed. Each experiment was run once, different for the original paper where the MNIST was run 4 times and the CIFAR 10 times, with different hyperparameters.

Sparsity and density of the network were two items continuously changed in order to see the effect on the network. The authors claimed that an increase on these is positively correlated with an increase in test accuracy. These claims are not verified by this reproducibility study as we did not tune this during the experiment.

| Name | Stable RNN? | Seq MNIST Best | PerSeq MNIST Best | Seq CIFAR Best |
|---|---|---|---|---|
| Sparse Combo Net (original paper) | Yes | 99.04% | 96.94% | 65.72% |
| ML Reproducibility Challenge | Yes | 99% | 96.73% | 64.48% |

Overall, our results were very close to the ones from the paper, but slightly lower as shown on the table above.

## MNIST

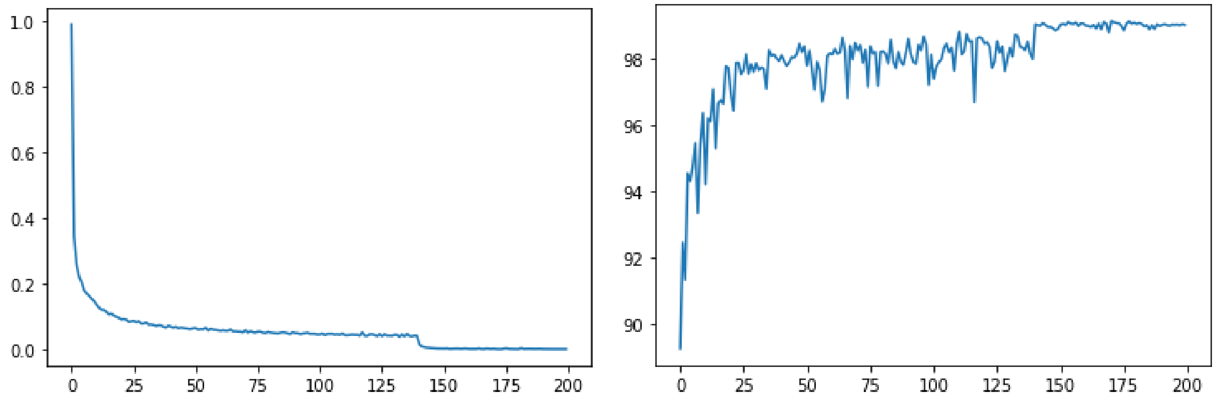

Figure 1.

## P-MNIST

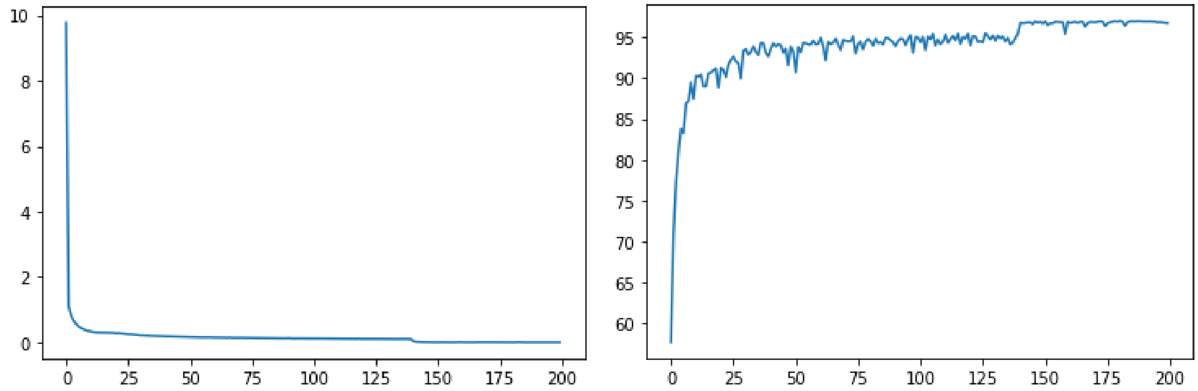

Figure 2.

## CIFAR-10

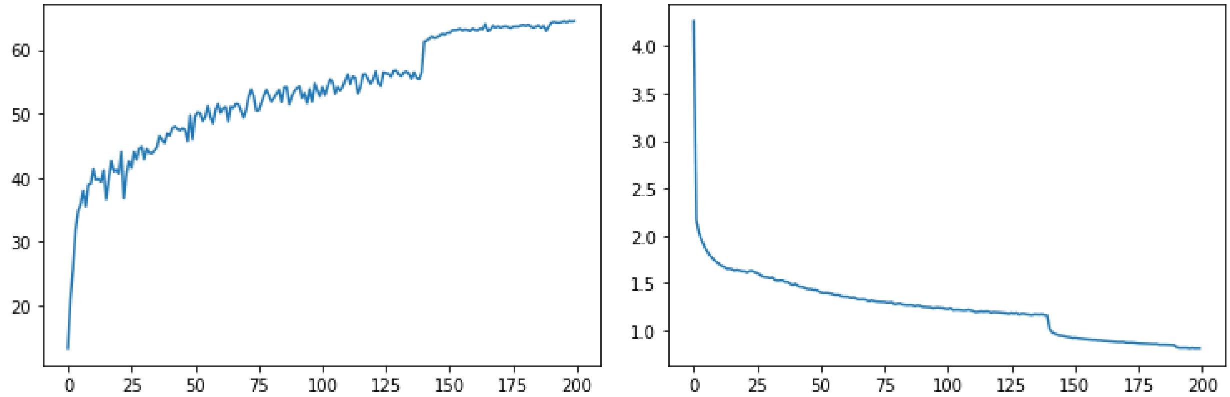

Figure 3. On the left, test accuracy metrics per epoch up to epoch #200. On the right, training loss is being recorded showing a decreasing pattern that still continues after the #150 epoch plataus before reaching epoch #200.

*Add comparison between how the accuracy and loss behaved in my experiment vs. in the paper*