DeepZip: Lossless Data Compression using Recurrent Neural Networks

Mohit Goyal*, **Kedar Tatwawadi**, **Shubham Chandak**, **Idoia Ochoa***
IIT Delhi, UIUC*, Stanford University

Introduction

There has been a tremendous surge in the amount of data generated. New types of data, such as **Genomic data, 3D-360 degree VR Data, Autonomous Driving Point Cloud data** are being generated. A lot of human effort is spent in analyzing the statistics of these new data formats for designing good compressors.

It is well known in information theory that good predictors form good compressors. Thus can neural networks be efficiently used for compression?

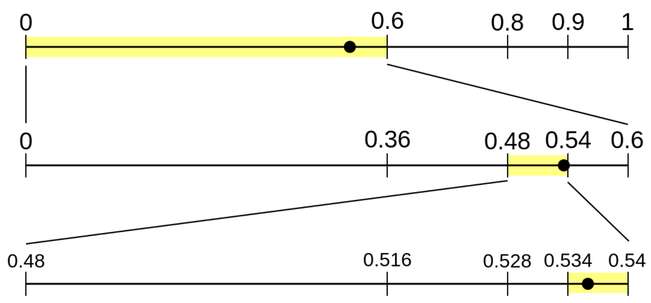
DeepZip

The DeepZip performs Lossless Compression and consists of:

1. **Deep Probability Estimator**: The Neural Network based block acts as a conditional probability estimator for the next symbol, given the past
2. **Arithmetic Coder**: The Arithmetic coding block uses the conditional probabilities for optimal compression/decompression:

$$\frac{1}{N}\log_2\frac{1}{\hat{p}(S^N)} \leq \bar{L}_{AE} \leq \frac{1}{N}\log_2\frac{1}{\hat{p}(S^N)} + \frac{2}{N}$$

Arithmetic Encoding of i.i.d. source

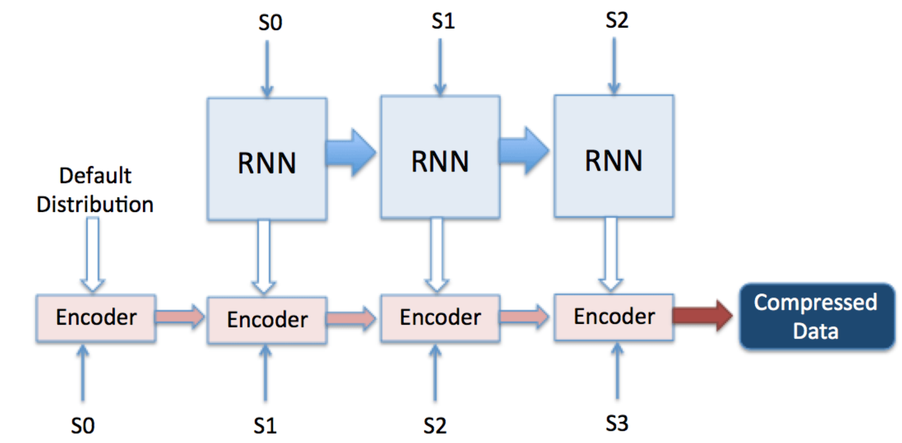


Model Framework

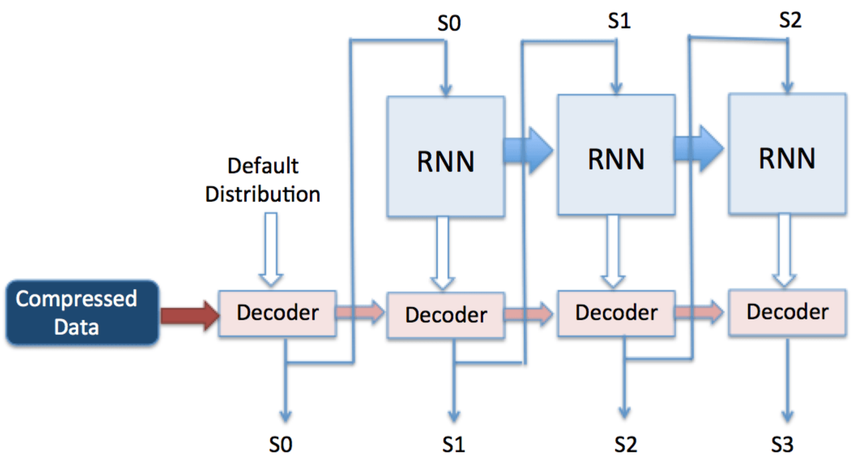
The Model framework is as follows:

1. **Distribution as output:** The RNN based model outputs a distribution over the alphabet set, which is the input to the Arithmetic Encoder.
2. **Arithmetic Coder State:** Arithmetic Encoder can be imagined as a FSM which keeps the `range` State, which is passed on as in the RNN.
3. **Causality of Input:** The RNN Estimator needs to be causal, and can have input features based only on the previously encoded symbols.
4. **Weight Update:** The weight update (if performed) should be performed in the exactly same way in the encoder and the decoder.

RNN-Arithmetic Encoder Framework



RNN-Arithmetic Decoder Framework



Experiments & Datasets

Datasets:
We experiment with the following synthetic datasets

1. **i.i.d. Sources**: We consider i.i.d. sources with a variety of parameters.
2. **Markov-k sources (XOR)**: Markov-k sources are 0-entropy sources with Markovity of k. They are governed by:
$$S_n = S_{n-1} + S_{n-k} \pmod M$$
Markov-k sources are difficult to compress, as they are a type of Pseudo-Random-Number-Generated sequences (Lagged Fibonacci PRNG).
3. **Hidden Markov Model (HMM):** We simulate a HMM source where the hidden state follows the Markov-k sequence described earlier.
$$S_{n+1} = X_n + X_{n-k} + Z_n \pmod M.$$
Here, the hidden process X_n is Markov-k, and Z_n is the added i.i.d. noise.

We also experimented with the following real datasets:

- **text8 Dataset:** Wikipedia text dataset.
- **HGP-Chr1**: The Human Genome Project DNA Chromosome-1 reference
- **C. elegans Genome**: *C. elegans* whole genome data
- **PhiX quality dataset**: Genomic quality value dataset from sequencing of PhiX virus

Results

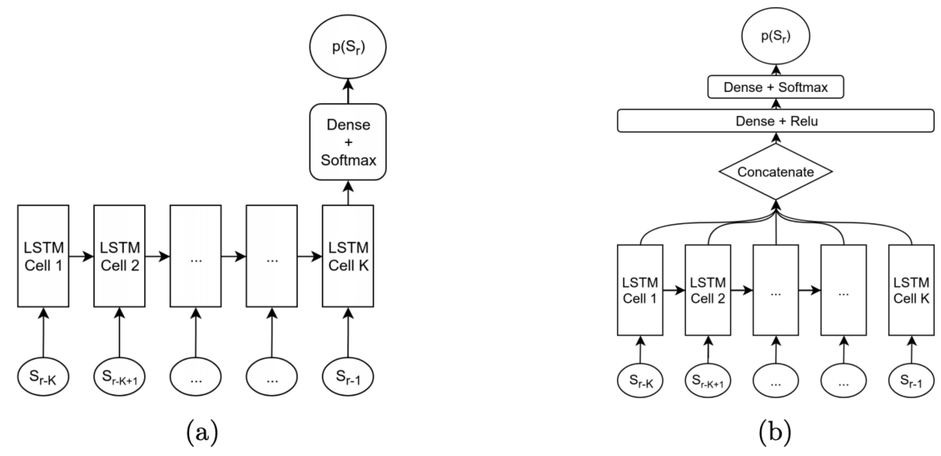


Fig 2: LSTM and LSTM-multi models

DeepZip on Synthetic datasets

Dataset	Seq. Length	Gzip	BSC	DeepZip		
				FC	biGRU	LSTM-multi
IID	10M	0.81	**0.60**	0.98	0.76	1.20
XOR20	10M	1.51	**0.06**	0.40	0.18	0.63
XOR30	10M	1.51	1.26	0.40	**0.18**	1.87
XOR40	10M	1.49	1.26	**0.40**	1.43	1.87
XOR50	10M	1.48	1.26	0.40	**0.18**	0.63
HMM20	10M	1.49	0.87	0.98	**0.76**	1.87
HMM30	10M	1.49	1.26	0.98	**0.76**	1.21
HMM40	10M	1.49	1.26	**0.98**	1.42	1.87

Fig. 5: synthetic datasets, XOr, HMM and IID

Comparison of DeepZip performance on real datasets

Dataset	Seq. Length	Gzip	BSC	DeepZip		
				FC	biGRU	LSTM-multi
H. chr1	249M	60.58	50.43	49.37	48.80	**48.56**
C. E. chr1	15M	4.03	**3.49**	3.81	3.58	4.02
C. E. genome	100M	26.97	23.38	23.41	**23.13**	23.41
text8	100M	33.05	**20.95**	25.49	23.37	26.71
PhiX Quality	100M	6.22	4.38	4.58	**4.35**	4.79

Dataset	FC		biGRU		LSTM-multi	
	Model	Sequence	Model	Sequence	Model	Sequence
H. chr1	0.39	48.98	0.17	48.62	0.62	47.95
C. E. chr1	0.39	3.42	0.17	3.40	0.62	3.98
C. E. genome	0.39	23.02	0.17	22.96	0.62	22.79
text8	0.40	25.09	1.74	21.63	0.63	26.09
PhiX Quality	0.39	4.19	0.17	4.18	0.62	4.18

Fig. 6: DeepZip on real text, genomic datasets

Code and additional details

Github Link: https://github.com/mohit1997/DeepZip
ArXiv Paper: https://arxiv.org/abs/1811.08162