Compression of Genomic Variants using Convolutional Autoencoders

Andrew Zhang and Kalyan Palepu
Mentor: Dr. Gil Alterovitz
7th Annual PRIMES Conference
May 22, 2017

Genomic Sequence Data

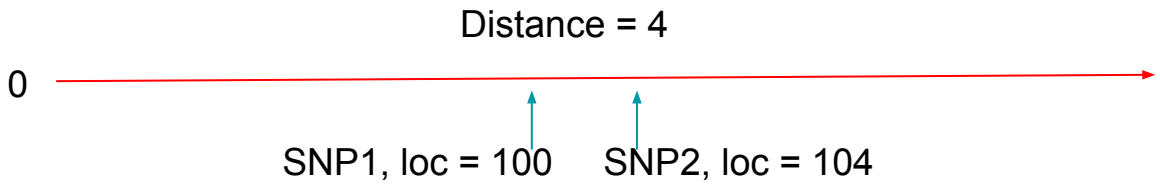
- Genome sequence data is key for disease diagnosis, and precision medicine
- With the development of High Throughput Sequencing (HTS) technology, more data is produced daily
  - An individual genome can be hundreds of gigabytes in size
- The massive amount of data is very difficult to store and process
  - This is why we need to compress the data!

Genomic Variants

- The basic idea behind genomic variants is that most genomes are very similar
  - Generally, less than 1% of the genome is different among different people
- Therefore, it is generally better to store the *differences* between an individual's genome and some reference genome
- VCF: Variant Call Format. It is a text file that stores only the variations with a reference genome.
  - Example: At position 1452, the reference was A, but this genome had C
  - At position 4214, the reference was T, but this genome had A
- **Our project compresses these VCF variant files**

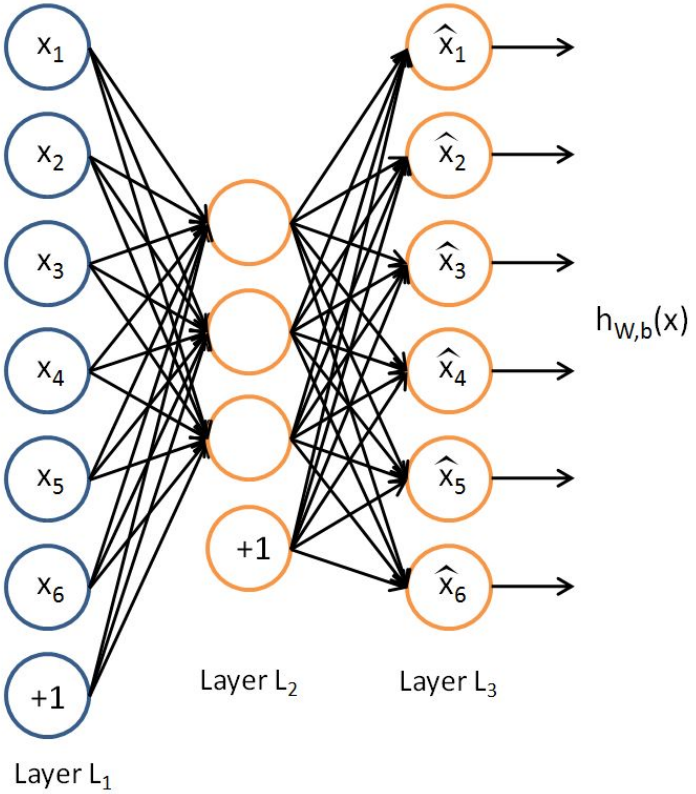
Past Research in Variant Compression

- DNAZip
  - Does not leverage biological patterns
  - Compresses VCF files by splitting the data into two sections:
    - Variants without position (e.g. A became G, then T became A). These are also known as SNPs
      - This part of the data was compressed referencing a database of common variant strings
    - Distances between positions (smaller than absolute positions)
      - We focused on compressing these distances further



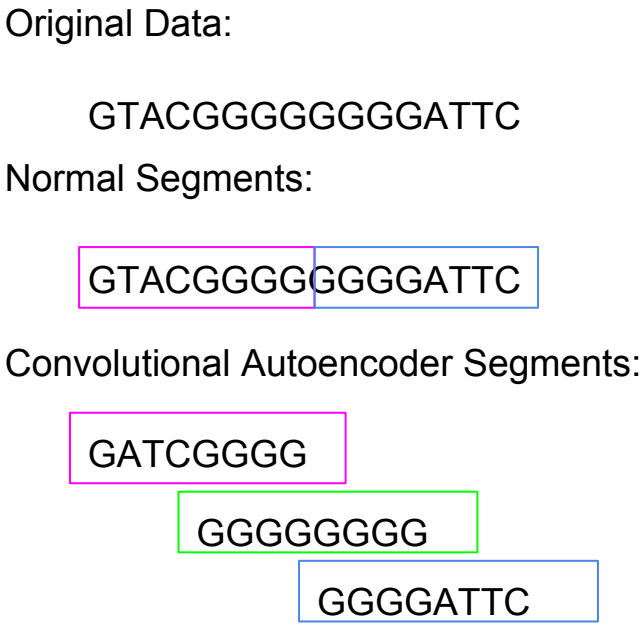
What is an Autoencoder?

- A form of neural network
  - Comprised of nodes which take some input and compute some output based on that input
- The autoencoder is trained to output the data which was input
- **Because the number of nodes in the middle is less than the number of inputs, the data is effectively compressed!**
  - The nodes before the smallest layer comprise the encoder network, and the nodes after the smallest layer comprise the decoder network



Convolutional Autoencoder

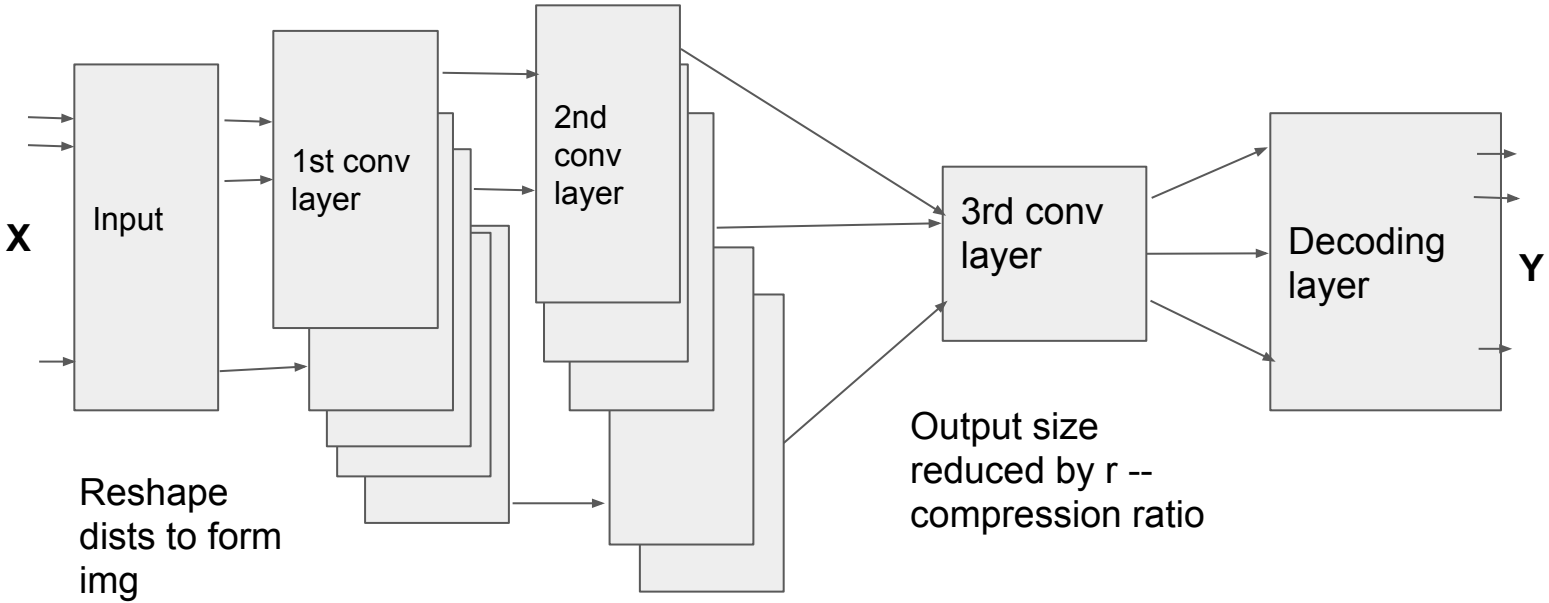
- A normal autoencoder has a fixed input size; to compress large strings of data, we split it into “segments,” and compress each segment individually
- A convolutional autoencoder, generally used for images, creates overlapping segments, to make sure we don’t skip important patterns.



Neural Network Graph used for Distance Compression

- For encoding, we used a three layer convolutional network on the encoding side for feature learning
  - Convolutional Autoencoder assumes inputs are images -- we convert distances to square images
  - Convolutional Autoencoder uses much smaller set of nodes per layer, so we can build deeper encoder
  - Convolutional Autoencoder proven to be able to learn image features in the literature -- will try to use it to learn biological features
- For decoding, we used a single layer
  - This makes decoding faster
  - Also minimize decoder parameter size

Conv NN graph -- continue



Errors

- No autoencoder is perfect, so we do get some errors when reproducing the data
  - We faithfully reproduce over 90% of the data
- However, genomic data compression *must* be lossless
  - Some diseases are the result of a single variant
- To make the autoencoder lossless, we record an “error matrix,” which stores all of the errors that the autoencoder has

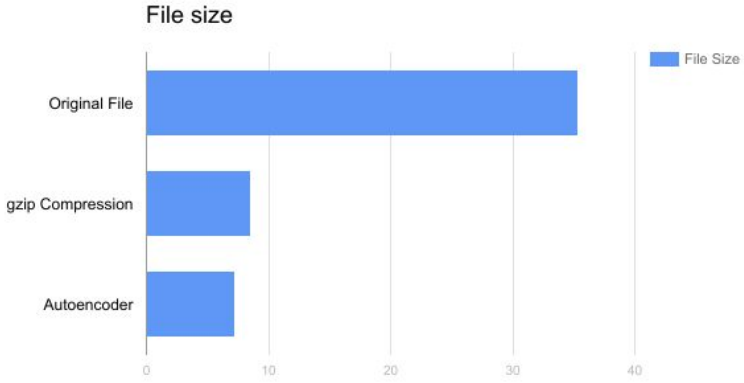
Test results with JW data

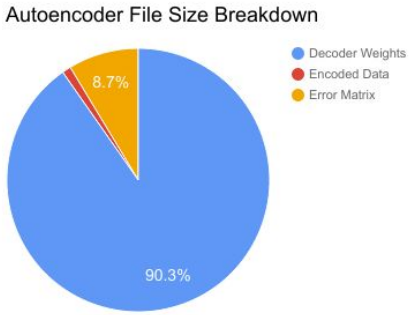
- Parsed file size: 35MB.

  Include locations parsed from JW VCF file

- GZIP performance: 8.47MB
- Autoencoder performance: 7.18MB
  - Decoder weight size: 6.48MB
  - Encoded data size: 0.078MB
  - Error matrix size: 0.622MB
  - Better than GZIP even with decoder matrix

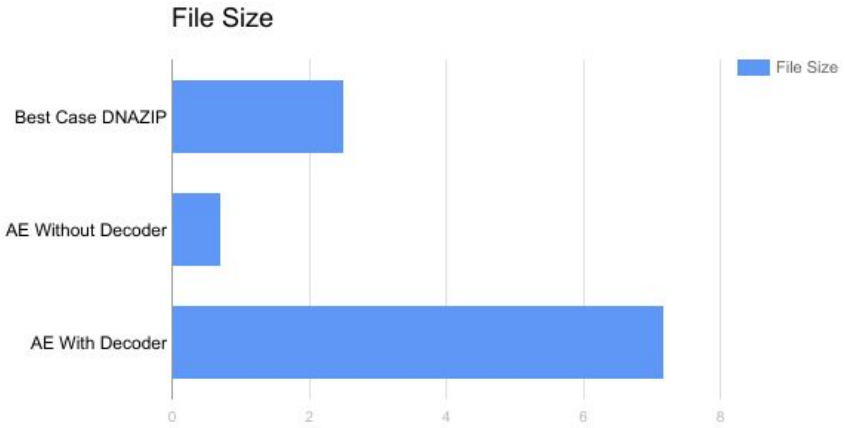
  - Almost 5x compression!





Results -- Conv Autoencoder compare with DNAZip

- Not better when including the decoder size, our goal is to generalize the decoder, so that we don't have to store the decoder for every file
  - This is likely to happen, given that we have had success sharing decoders for multiple individual files
- Autoencoder compressed data alone: 0.72MB
- DNAZip (best case estimate): 2.5MB
- Convolutional Autoencoder w/o decoder weights is 3.5x better than DNAZip



Future Work

- Have compressed 77% of the distances, ~2.5 millions, using autoencoder. We will work on to cover ~90%; compress the rest using statistical methods
- As stated before, we are working on creating a universal decoder, so that we do not need to store a decoder for each file.

Conclusion

- Built a convolutional autoencoder to compress 77% of all distances
- Tested with James Watson’s sequenced data in VCF format
- Compression ratio better than GZIP, and better than DNAZip if excluding decoding matrix

Acknowledgements

- Dr. Gil Alterovitz and Maksym Korablyov
- MIT PRIMES
- Our Parents
- Andrew Gritsevskiy and Adithya Vellal