RNN Convolutional Code Decoder

Calvin Higgins, Justin Watkins, Tuyetlinh Nguyen

University of Rhode Island

December 21, 2022

Overview

Table of Contents

Error Correcting Codes

In modern digital communications, error control coding is nearly ubiquitous, from CDs, flash drives, phone calls, TV, and anything transmitted over the internet.

Definition

Channels are the medium through which data is transmitted between devices. Channels can introduce errors into the data during transmission.

Definition

Error correcting codes add redundancy to data prior to transmission to ensure that errors can be corrected based on the received data.

Convolutional Codes

Convolutional codes are a powerful class of error correcting codes that use short memory and convolution operators to sequentially create coded bits.

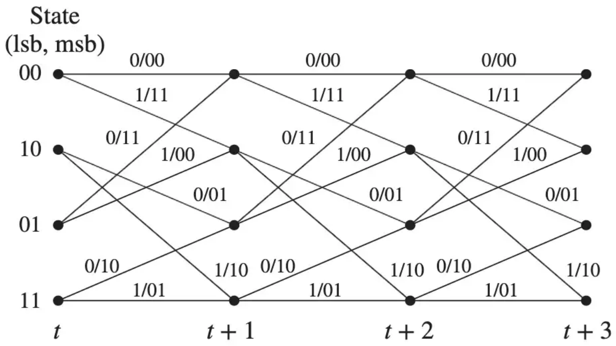


Figure: Trellis representation

The Viterbi Decoder

Goal

Determine the **most likely** sequence of states that could have produced the given sequence of received bits

Using the trellis representation of the encoding process, the Viterbi decoder determines the **most likely path** along this trellis.

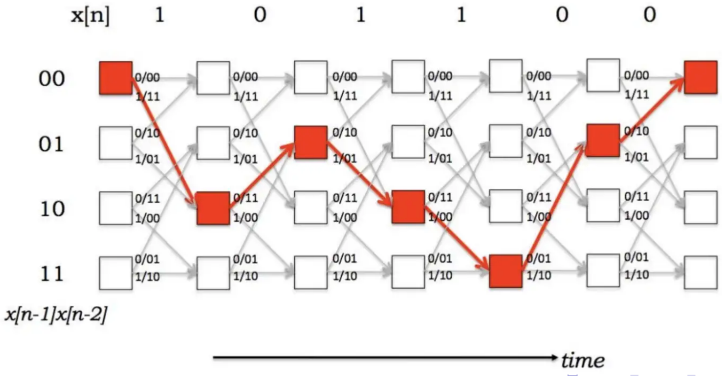


Table of Contents

Goal of the Project

In 1996, Wang and Wicker determined that artificial neural networks with hand-picked coefficients can reproduce the optimal Viterbi decoder.

Our Goal

1. Implement a convolutional code decoder using neural networks in an attempt to **learn** this decoder in a data-driven manner.
2. Compare its reliability against that of the Viterbi algorithm.

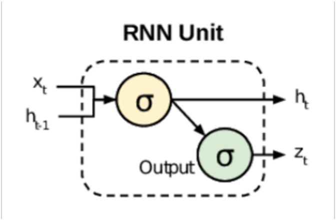
Note: This algorithm relies heavily on assumptions of the channel. However, practical channels will most likely break the fundamental assumption behind the algorithm.

On such channels, the Viterbi algorithm is suboptimal, giving an opportunity for a neural network to have better performance.

Table of Contents

Recurrent Neural Networks

We want to map encoded data (potentially with errors) to decoded data. This is called a **sequence-to-sequence** problem.

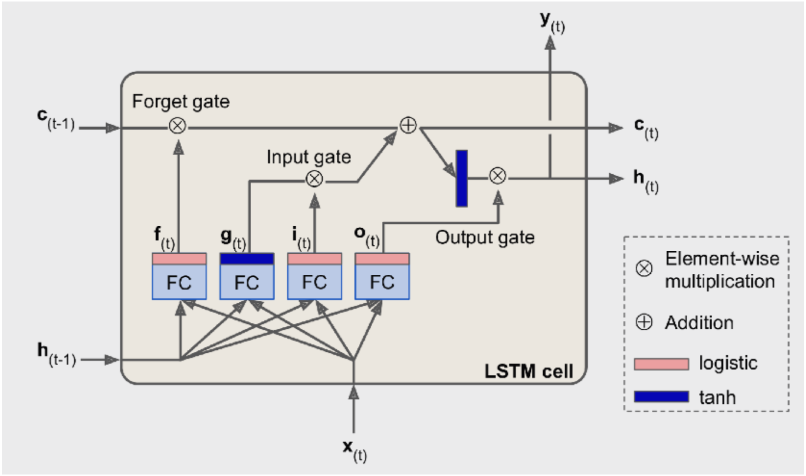


.

Recurrent neural networks (RNNs) are a type of neural network equipped with “memory” that enables them to learn from sequential data. RNNs excell at sequence-to-sequence problems!

LSTMs

LSTMs are powerful RNN variant with longer term memory. Practically speaking, they are a drop-in replacement with a lot of kick!



Our Model

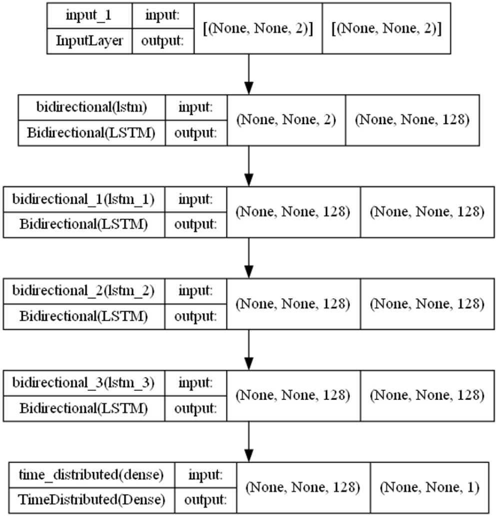


Figure: The Structure of Our Model

Data and Baseline

We generated our own dataset by emulating the process of encoding, transmitting and decoding bit sequences.

We generated bit strings of length 64 and pushed each through a convolutional encoder and "transmitted" them through a simulated channel. This channel introduced burst errors of a certain length with probability 0.04.

In order to have a baseline to analyze the performance of our model, we also ran an implementation of the Viterbi decoder to provide a concrete comparison to our model.

Training

When training the models:

- For each burst length in the range [1, 31], we trained a separate model with a Nadam optimizer with an initial learning rate of 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$.
- We used the MSE loss function.
- When the loss did not improve for more than 5 epochs, the learning rate was reduced by a factor of 10.
- If the loss did not improve for more than 10 epochs, training was halted and the model with the lowest achieved loss was selected.

Table of Contents

Comparison Against the Viterbi Decoder

We evaluated each final model on 2048 new samples to generate final loss and Hamming distance metrics.

Additionally, we evaluated a Viterbi decoder with a decision depth of 15 and AWGN assumptions on the same samples to serve as a baseline.

The following graph was produced comparing the Hamming distance between each original sequence and what the model and Viterbi decoder predicted for that sequence.

Results

In the following graph, the average Hamming distance for each burst length is plotted as a single point and the error bars represent values within one standard deviation.

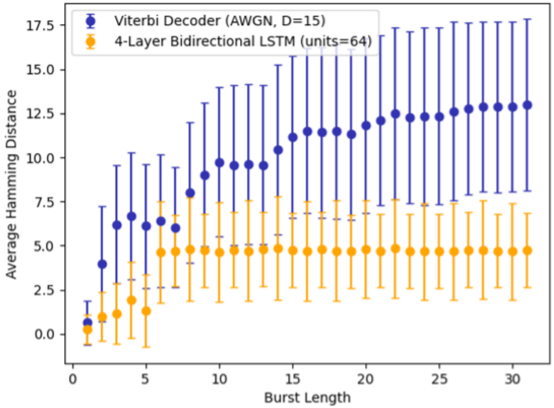


Table of Contents

References

Yair Mazal (2021)
Intro to Convolutional Coding — Part I & Part II
Medium - Nerd for Tech https://medium.com/nerd-for-tech/into-to-convolutional-coding-part-i-d63decab56a0

Hyeji Kim and Sewoong Oh (2020)
Decoding convolutional codes
Inventing Codes via Machine Learning
https://deepcomm.github.io/jekyll/pixyll/2020/02/01/learning-viterbi/.

Todd K. Moon (2005)
Chapter 12: Convolutional Codes
Error Correction Coding: Mathematical Methods and Algorithms, 452 – 525.

Questions?