RNN Convolutional Code Decoder

Calvin Higgins, Justin Watkins, Tuyetlinh Nguyen

University of Rhode Island

December 21, 2022

Overview

Table of Contents

Convolutional Codes

Convolutional codes are a powerful method of encoding messages, by using short memory and convolution operators to sequentially create coded bits.

Definition

A **linear shift register (LSR)** is a shift register whose input bit is a linear function of its previous state.

The most commonly used linear function of single bits XOR.

A convolutional encoder utilizes linear shift registers to encode k input bits into n output bits, thus yielding a code of **rate** $R = \frac{k}{n}$. Each output bit depends on the previous L input bits, where L is called the **constraint length**.

Convolutional Encoding

This encoder depicted in the figure below corresponds to the following **generator polynomials**:

$$\begin{aligned} g^{(1)} &= 1 + x^2 \qquad g^{(2)} = 1 + x + x^2 \\ g^{(1)} &= [1, 0, 1] \qquad g^{(2)} = [1, 1, 1] \end{aligned} \tag{1}$$

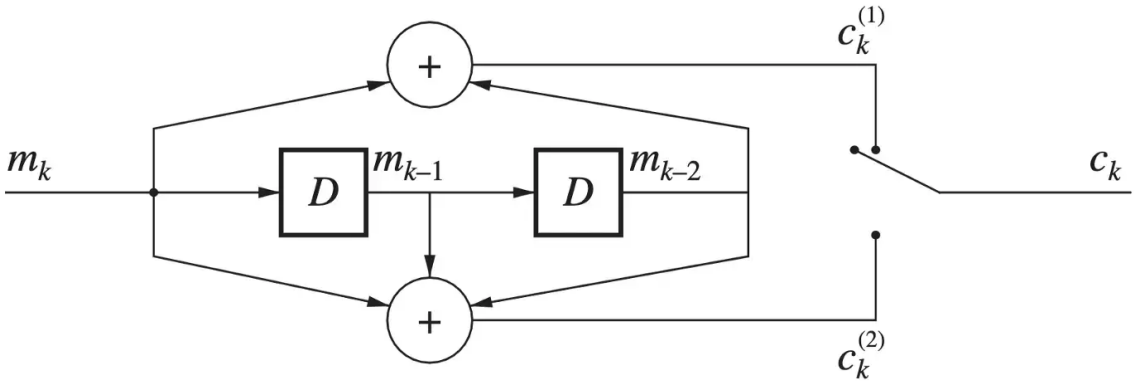


Figure: Convolutional encoder with rate $R = \frac{1}{2}$ and constraint length $K = 2$

Finite State Machine

Other representations of this encoder are a **finite state machine** and a **trellis**.

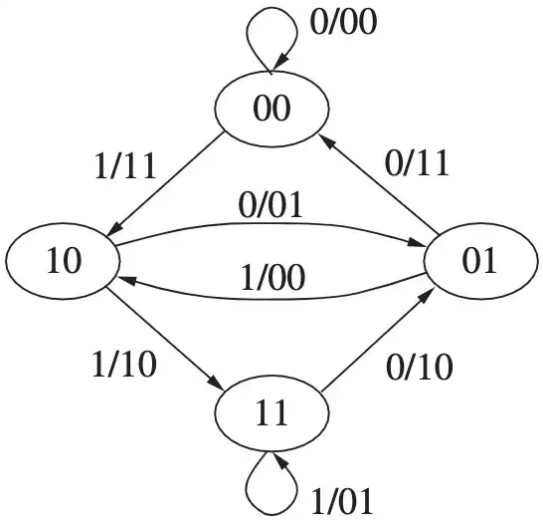


Figure: FSM representation

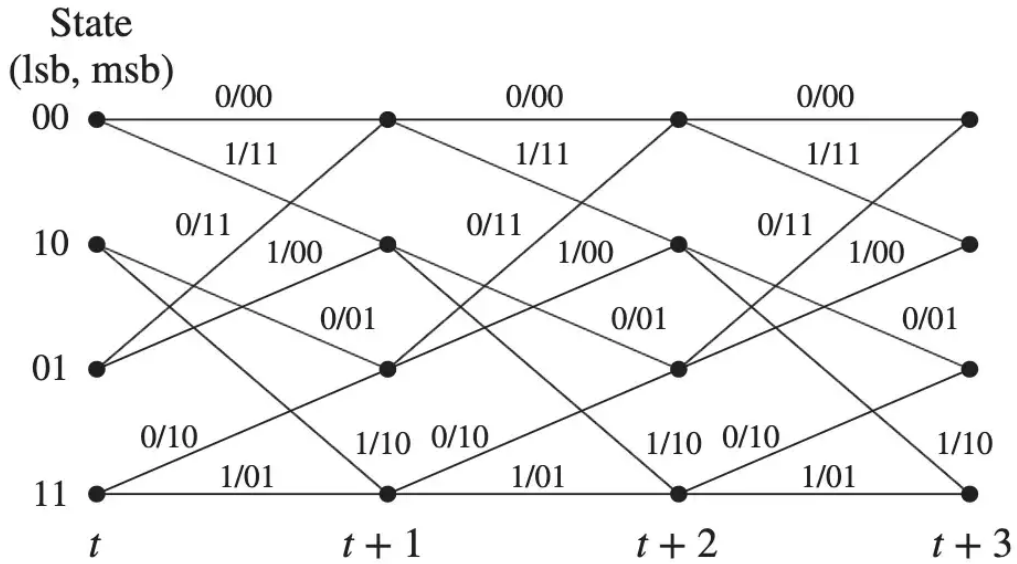


Figure: Trellis representation

The single digit on each edge indicates the input bit and the two digits indicate the output according to the generator polynomials.

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.

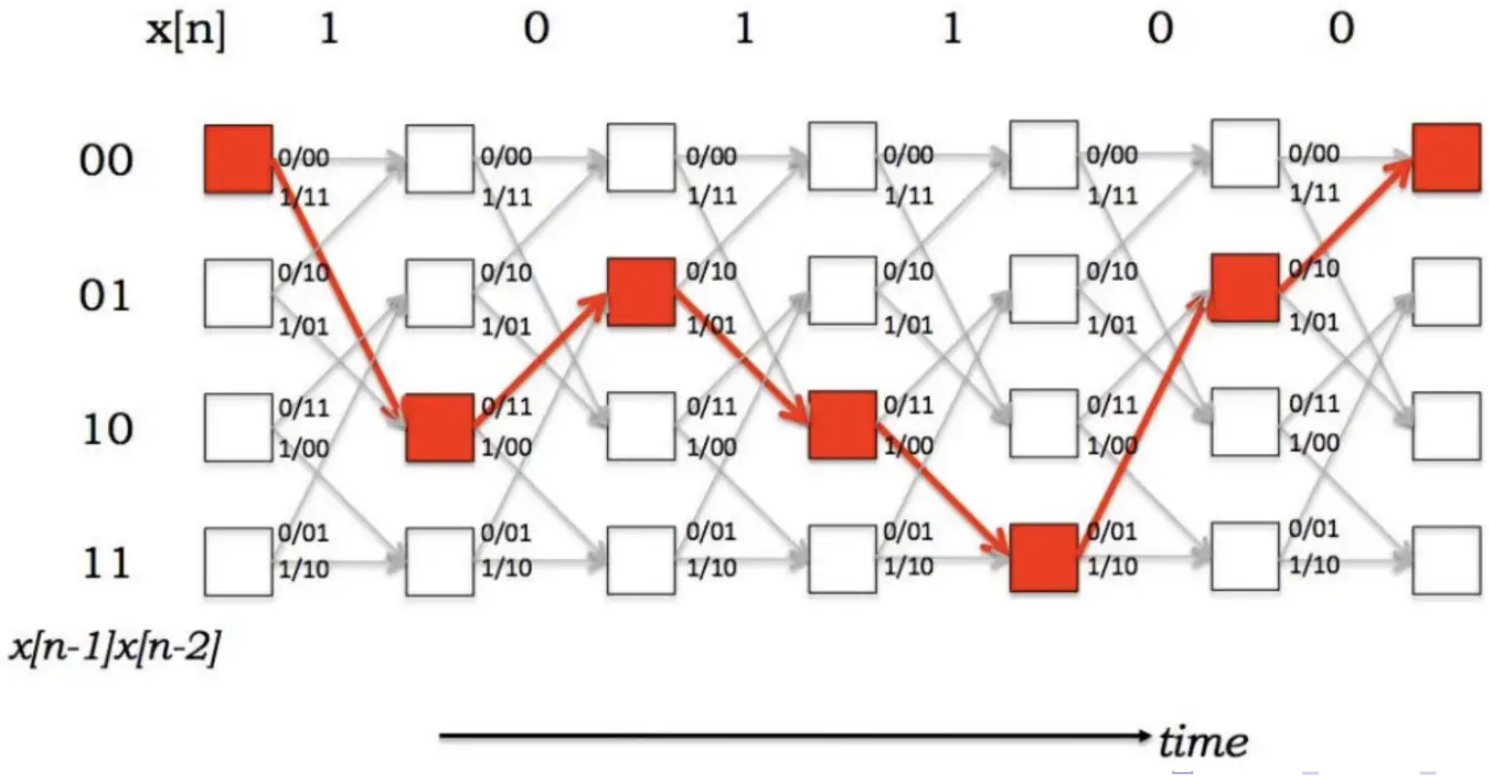


Table of Contents

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.

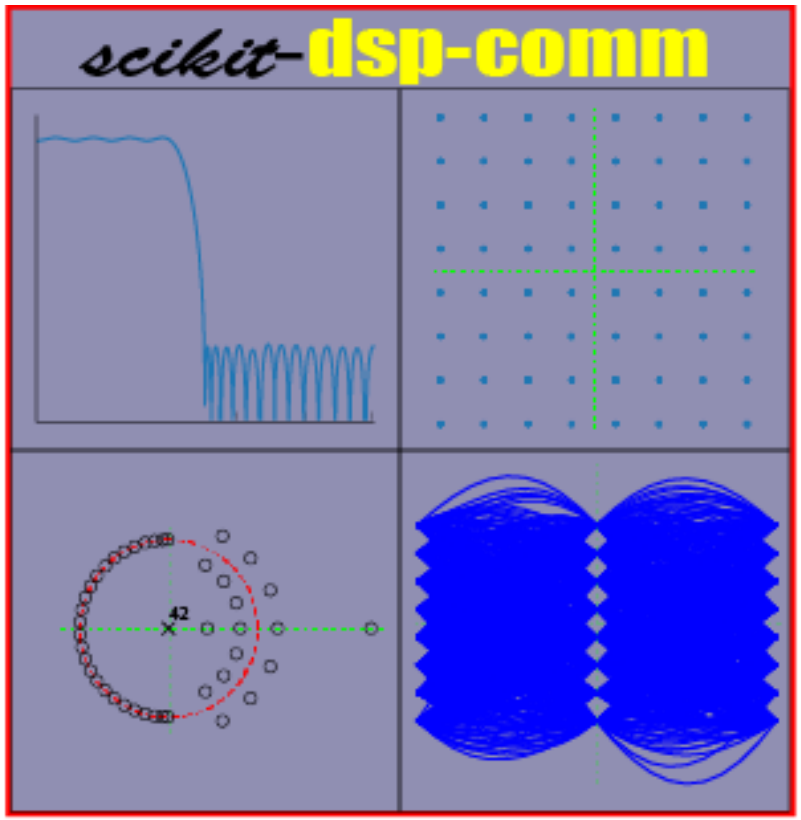
On such channels, the Viterbi algorithm is suboptimal, giving an opportunity for a neural network to have better performance.

Table of Contents

scikit-dsp-comm

The scikit-dsp-comm package "is a collection of functions and classes to support signal processing and communications theory teaching and research."

Due to some discrepancies between our model and the Viterbi decoder from the package, a patch was created to ensure both outputs were comparable.



Tensorflow with Keras

The Keras package is a high-level wrapper over Tensorflow 2 for implementing neural networks.

We used Keras to implement both our models and our training pipeline. In particular, we made extensive use of the LSTM layer for the model implementation and the Keras Sequence API for dataset generation.

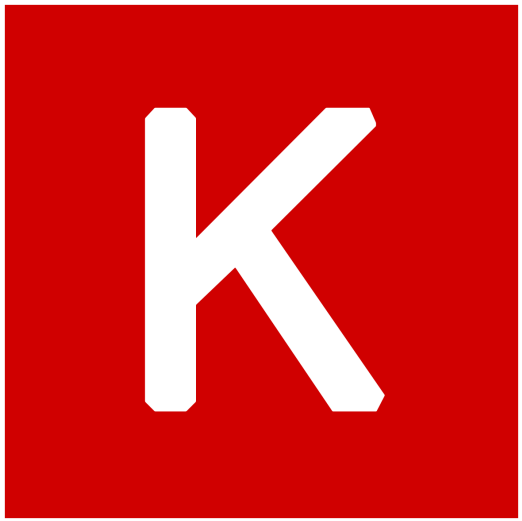


Table of Contents

Our Model

Our model consisted of 4 stacked layers of Bidirectional-Long Short-Term Memories (LSTMs) with 64 units per layer and the tanh activation function and an output layer being a single time distributed neuron with the sigmoid activation function.

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.

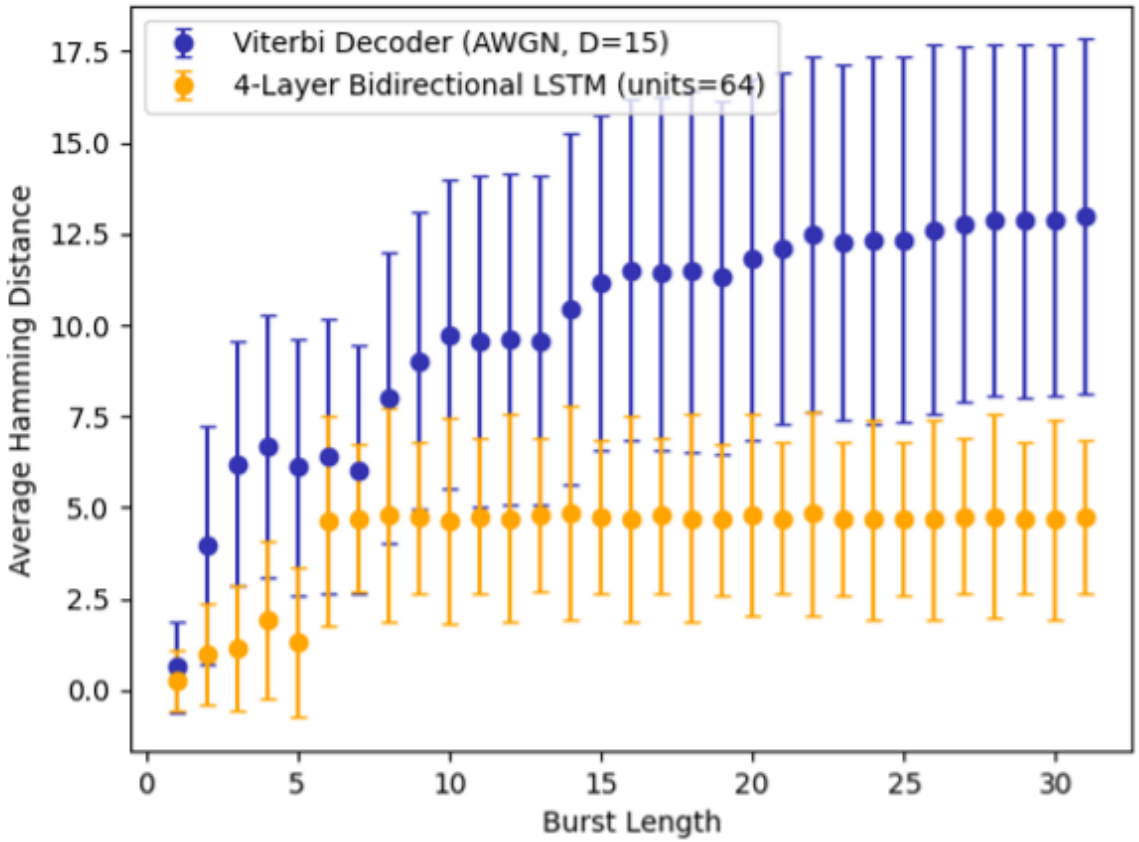


Table of Contents

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?