CÓDIGOS CONVOLUCIONAIS CORRETORES DE ERROS

Gabriel Vinícius Brandão (PIBIC/CNPq/FA/Uem), Eduardo Brandani da Silva (Orientador), e-mail: ebsilva@uem.br

Universidade Estadual de Maringá / Centro de Ciências Exatas
Maringá, PR

Matemática - Matemática Aplicada

Palavras-chave: códigos corretores de erros, códigos convolucionais, teoria da informação.

Resumo:

Através do estudo da fundamentação necessária para construção de códigos convolucionais corretores de erros, o projeto teve como foco compreender a estruturação de um código convolucional, isto é, o estudo dos codificadores convolucionais, noções de distância e decodificação.

Introdução

A transmissão de informações em comunicações digitais se apresenta como um problema muito atual, assim demandando novas propostas de sistemas de transmissão de dados, bem como sistemas de codificação/criptografia. Um sistema simples de comunicação basicamente consiste de um sistema que gera, envia e recebe dígitos binários, ou seja, *bits*. Em transmitindo estes bits através de um canal, ou guardando-os em uma memória, é provável que, devido ao ruído inerente à comunicação, alguns dos bits sejam recebidos de forma diferente à enviada. Assim, é de grande interesse que recuperemos as informações que sofreram mudanças durante a sua transmissão. Dessa maneira, os códigos corretores de erros apresentam uma forma bem estruturada de adicionar redundâncias às informações e, posteriormente ao seu envio por um canal de transmissão, decodificá-las de volta a um estado legível.
Os códigos convolucionais corretores de erros, por outro lado, são um tipo de códigos corretores de erros que, introduzidos por Peter Elias em 1955, apresentam uma forma diferente dos mais tradicionais códigos de bloco (da Teoria de Claude Shannon) para codificação e decodificação de informações, ainda que muito da estrutura e fundamentos são reutilizados por este novo olhar. Apesar da teoria dos códigos de blocos se apresentar muito mais sólida e rica matematicamente do que a teoria dos códigos convolucionais, existem diversos fatores para a utilização da segunda. Um bem importante é a sua capacidade de decodificar informações utilizando as

chamadas *soft-decisions* com mais facilidade do que a maioria dos códigos mais tradicionais.

Revisão da Literatura

O estudo deu-se por meio da leitura de livros e artigos relacionados ao assunto, bem como a apresentação de seminários semanais com o orientador, e leituras aprofundadas em encontros. A metodologia do projeto foi a típica da área da Matemática, estudando as estruturas referentes à construção dos códigos convolucionais corretores de erros.

Resultados e Discussão

Abaixo vamos relacionar as principais definições e conceitos estudados, bem como os resultados mais importantes que foram estudados. Maiores detalhes podem ser encontrados nas referências listadas ao fim do resumo.
Como comentado, num código corretor de erros, buscamos codificar as informações, enviá-las através de um canal, de forma que a codificação forneça uma ferramenta para decifrar a mensagem que chega na recepção. Iniciaremos por definir um código e um codificador convolucionais.

Tomemos um espaço finito, normalmente este sendo o $\square_2$ = {0, 1}, o espaço binário. Assim, um código convolucional é pensado como um código linear sobre este espaço, contudo, os bits de informações por ele codificado não são organizado em blocos, mas sim, formam uma sequência infinita e dependente do tempo. Dizemos que um código convolucional é linear pois dada uma sequência infinita de entrada $\square = \square_0\square_1\ldots$ o codificador irá, em dependendo de quantas funções lineares a saída está submetida, entrelaçar e fazer operações com $\square$, resultando na saída $\square$. Essas operações são todas lineares em $\square_2$. Tomemos como exemplo o codificador convolucional da Figura (1). Neste exemplo, as funções as quais estão submetidas as saídas são respectivamente $\square_1 = (11100\ldots)$ e $\square_2 = (10100\ldots)$. Essas funções podem ser obtidas diretamente da figura (realização) do codificador, onde a função $\square_1$ e $\square_2$representam as saídas na primeira e segunda "linhas" (observe os sinais de $\oplus$). Neste codificador em específico, temos uma taxa $\square = 1/2$, uma vez que para cada bit de entrada, dois bits de saída são emitidos através do canal. O registrador, neste caso de **memória** ou **comprimento** 2 são as "caixas" (memórias) por onde os bits de entrada são "empurrados" para se formarem as saídas, que podem ser vistas como uma convolução da entrada com as funções $\square_1$e $\square_2$.

Dessa forma, a saída pode ser entendida como:

$$\square = (\square_0\square^1\square\square_0\square^2\square\square_1\square^1\square\square_1\square^2\square\ldots) = \square(\square) = \square(\square_0\square_1\ldots)$$

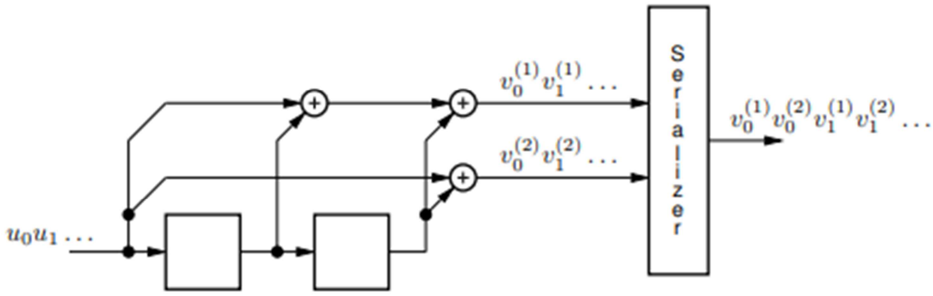


Figura 1 - Exemplo de um codificador convolucional. Fonte: (Johannesson; Zingangirov, 2015).

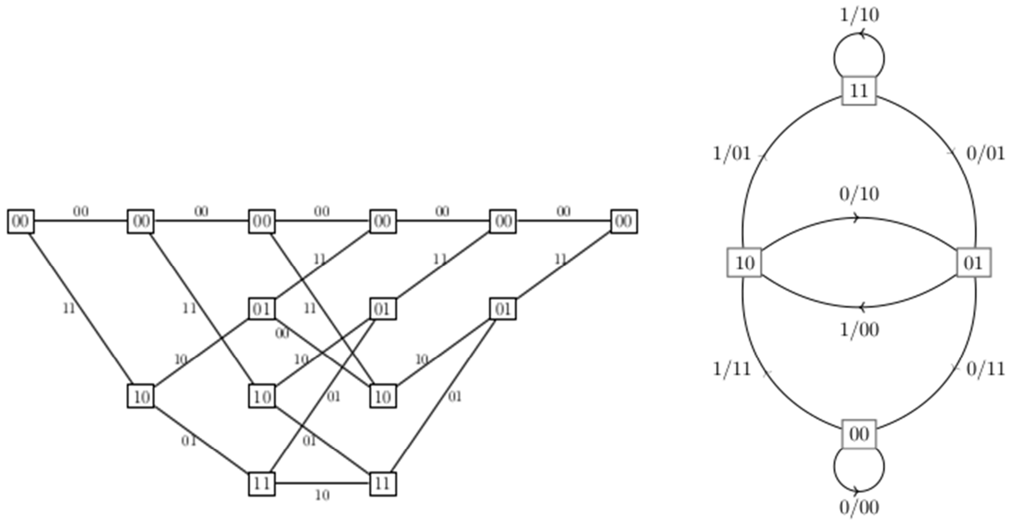
Facilitamos a escrita das funções com a escrita matricial □= □□, onde □ é chamada **matriz geradora.** Contudo, uma vez que as funções são infinitas e dependentes do tempo, com o intuito de simplificar a escrita, utilizamos a transformada-D. Explicaremos do que se trata essa transformada com o seguinte simples exemplo. Seja □ = (*110101*) então no domínio da transformada-D,

□ = *1* + □ + □³ + □⁵.

Assim, a matriz □que seria tratada como uma matriz infinita agora fica escrita em termos da indeterminada D e é tomada como □(□). Dessa forma, ainda utilizando o exemplo da Figura 2, a matriz geradora □(□) é dada por□(□) = (*1* + □ + □²   *1* + □²).

Uma outra importante definição tangendo o estudo é a de estados. O estado de um sistema é uma descrição de sua história passada, na qual, juntamente com a descrição das entradas presentes e futuras, são suficientes para determinar as saídas presentes e futuras. Num codificador convolucional, o estado σ é determinado como o conteúdo de suas memórias (“caixas”). Assim, no exemplo da Figura (1), temos 4 possíveis estados: 00, 01, 10 e 11. É usual que desenhemos um chamado **diagrama de transição de estados**. Este diagrama para a Figura (1) está mostrado na Figura (2.b). Este diagrama indica além da transição dos estados, as entradas e as respectivas saídas do codificador. Assim sendo, dada uma entrada, por exemplo □ = *11010*, temos que a palavra codificada que será enviada pelo canal é □ = *11 01 01 00 10*.

Uma vez enviada pelo canal, a palavra provavelmente sofrerá distorções, e para isso, Viterbi inventou um simples, porém poderoso, algoritmo para a decodificação das informações. O **algoritmo de Viterbi** explora a estrutura de uma treliça (Figura (2.a)) para performar uma decodificação com máxima verossimilhança, isto é, de forma a reduzir a distância de Hamming entre as palavras recebida e enviada. De forma simples, o algoritmo conta os dígitos que foram recebidos errados eliminam-se os caminhos com mais erros (pois estes não poderiam possivelmente ser os caminhos corretos). Portanto, ao chegar ao fim da treliça, há somente um caminho sobrevivente, e com mínima distância de Hamming entre alguma palavra do código. Esta palavra é então escolhida como a palavra enviada. Um exemplo de decodificação pode ser encontrado na referência (Johannesson; Zingangirov, 2015, p. 33).



(a) (b)

Figura 2 - (a): Diagrama de treliça para o codificador da Figura (1). **(b):** Diagrama de transição de estados para o codificador da Figura (1). Fonte própria.

Conclusões

Os tópicos estudados sobre os códigos convolucionais e abrangendo a teoria da informação foram de muita utilidade para uma aproximação com a aplicação de uma teoria matemática dentro de um contexto tecnológico e aplicável. Levantaram-se várias discussões sobre demonstrações dos teoremas que fundamentam a teoria, possibilitando assim a introdução ao meio da pesquisa científica em uma área não abordada na grade curricular do seu curso.

Agradecimentos

Agradecimentos ao CNPq, à Fundação Araucária e à UEM pelo apoio financeiro.

Referências

Johannesson, R.; Zigangirov, K. S. Fundamentals of Convolutional Coding. 2. ed. [S.l.]: IEEE Press, 2015.