گزارش کار پروژه دوم درس انتقال داده

نوید اکبری

۸۱۰۸۹۵۰۲۳

برای پیاده سازی این تمرین کامپیوتری از زبان python استفاده شده است. برای بخش اول یعنی کدینگ منبع با استفاده از الگوریتم Huffman در ابتدا باید درختی را پیدا کنیم که با توجه به احتمال‌های داده شده بهترین کدینگ ممکن را انجام دهد. برای این کار از تابع generateHuffmanDictionary استفاده شده است. در این تابع ابتدا به تعداد حروفی که داریم Node درست می‌کنیم. در این Node ها مقدار فرکانس هر حرف و خود حرف وجود دارند. حال الگوریتم به این صورت می‌باشد که تا وقتی که تعداد نودهای نودهای موجود بیشتر از یک باشد در هر مرحله ۲ تا نود با کمترین احتمال را گرفته و با آن‌ها یک نود پدر درست می‌کند که احتمال آن برابر جمع احتمالات فرزندانش می‌باشد و مقدار الفبا آن هم برابر جمع دو الفبا فرزندانش می‌باشد. حال نودهای فرزندان را حذف می‌کنیم و نود پدر را اضافه می‌کنیم. تا زمانی این کار را ادامه می‌دهیم که فقط یک نود به عنوان root باقی بماند. حال به صورت برعکس درخت تولید شده را پیمایش می‌کنیم و کدها را درست می‌کنیم. پس از اجرای این کد با احتمال‌های داده شده کدهای زیر ساخته شده‌اند.

't': '111', 'f': '11011', 'v': '110101', 'q': '110100111', 'z': '110100110', 'x': '110100101', 'j': '110100100', 'k': '1101000', 'w': '11001', 'm': '11000', 'u': '10111', 'c': '10110', 'r': '1010', 'h': '1001', 's': '1000', 'e': '011', 'n': '0101', 'i': '0100', 'b': '001111', 'p': '001110', 'y': '001101', 'g': '001100', 'o': '0010', 'a': '0001', 'l': '00001', 'd': '00000'

حال که دیکشنری را داریم می‌توانیم متن داده شده را توسط تابع huffmanEncoding انکد کنیم. این تابع فقط با داشتن Code book به ازای هر حرف مقدار باینری جایگزین آن را برمی‌گرداند. به عنوان مثال برای متن آزمایشی "test" داده‌ی باینری 1110111000111 را برمی‌گرداند که می‌توان به راحتی با دیکشنری بالا به دستش آورد. حال برای عمل برعکس این کار یعنی دیکد کردن متن باینری داده شده از تابع huffmanDecoding استفاده شده است. برای این کار چون می‌دانیم که کدها یکتا هستن و به صورت prefixای از هم نیستند از ابتدا داده باینری شروع می‌کنیم و به جلو می‌رویم هرجا که به کدی رسیدیم که در دیکشنریمان بود مقدار الفبا آن را جایگزین می‌کنیم. با این کار متن اولیه‌ای که داشتیم ساخته می‌شود.

پس از این مرحله به مرحله کدینگ کانال می‌رسیم که توسط کدینگ Convolutional انجام می‌شود. این کار توسط تابع channelCoding انجام می‌شود. به دلیل این که یک استیت ماشین ثابت داریم می‌توانیم این کار را با استفاده از

گزارش کار پروژه دوم درس انتقال داده

نوید اکبری

۸۱۰۸۹۵۰۲۳

یک دیکشنری انجام دهیم. بنابراین یک دیکشنری با توجه به ماشین حالت داده شده ساخته‌ایم و کدهای معادل هر بیت را به عنوان خروجی می‌دهیم. با توجه به ماشین حالت داده شده به ازای هر بیت ۲ بیت خروجی می‌دهیم که طول فایل باینری وارد شده را ۲ برابر می‌کند این کار. به عنوان مثال رشته ورودی باینری 1110111000111 تبدیل به رشته‌ی 11001001010010011000110010 می‌شود. مرحله نهایی ساختن رشته باینری اصلی از دیکد شده آن توسط کدینگ Convolutional می‌باشد برای این کار از الگوریتم Viterbi استفاده می‌کنیم که بر اساس برنامه‌نویسی پویا می‌باشد. این کار توسط تابع channelDecoding انجام می‌شود. در اینجا نیز چون ماشین حالتی که برای Trellis داریم ثابت می‌باشد از یک دیکشنری استفاده کرده‌ایم و ۴ حالت مختلف را در نظر گرفته‌ایم. برای هر یک از این حالت‌ها یک مقدار Path Metric وجود دارد که باید به اندازه طول داده باینری آن را حساب کرده، یعنی برای بار اول باید به صورت forward حرکت کرده و همه Path Metric ها را حساب کنیم تا بتوانیم در مرحله بعد با برعکس آمدن این مسیر بهترین جواب را پیدا کنیم. در ابتدا مقدا Path Metric استیت 00 برابر صفر می‌باشد در حالی که بقیه استیت‌ها برابر بی‌نهایت، علت ای کار ماشین حالتمان در زمان کد کردن داده می‌باشد. در زمان کد کردن ما از استیت 00 شروع کردیم و با توجه به ماشین حالت میبینم که فقط ۲ حالت 00 و 11 امکان پذیر می‌باشد از این استیت بنابراین همواره بیت‌های اولیه یا 00 هستند یا 11 و چون از استیت 00 در Trellis می‌توان به این اعداد رسید از این حالت شروع می‌کنیم. با حساب کردن همه Path Metric ها مرحله Backward شروع می‌شود و بهترین مسیر موجود را می سازیم تا به جواب برسیم.

گزارش کار پروژه دوم درس انتقال داده
نوید اکبری
۸۱۰۸۹۵۰۲۳

بخش پایانی:

حالت تشخیص درست

Plain Text : navidakbari

Encoded Text : 0101000111010101000000000011101000001111000110100100

Channel Encoded Text :
0011110111100011001001011101110111100000000000000000011001001011110000000
1100101001100011000101111011111 0

Received Text :
0010110111100011001000010101110110100000000010000001100100101101000000 1
1000001001100011000101111011111 0

Channel Decoded Text : 0101000111010101000000000011101000001111000110100100

Decoded text : navidakbari

حالت تشخیص غلط

Plain Text : navidakbari

Encoded Text : 0101000111010101000000000011101000001111000110100100

Channel Encoded Text :
0011110111100011001001011101110111100000000000000000011001001011110000000
1100101001100011000101111011111 0

Received Text :
0011110011110011001001011101110010101000000000000000011001101010110000000
1100101001100001100111111011111 0

Channel Decoded Text : 0101000111010101000000000011011000001111011000100100

Decoded text : navidaclteao