به نام خدا

انتقال داده

تمرین کامپیوتری ۲

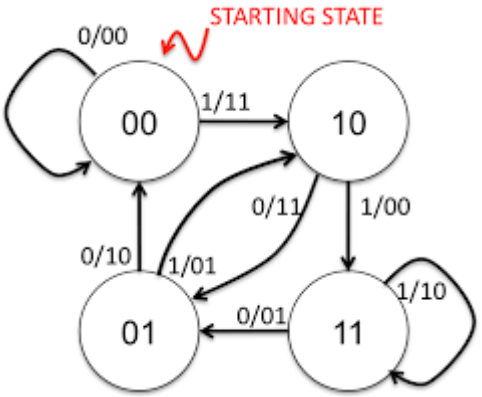
کدینگ منبع و کدینگ شبکه

کدینگ منبع

در بخش اول پروژه، قصد داریم به پیاده‌سازی یکی از روش‌های معروف کدینگ منبع بپردازیم. همان‌طور که در درس آشنا شدید، الگوریتم Huffman یک الگوریتم حریصانه برای پیاده‌سازی کدینگ منبع است. در این بخش، نیاز است که تابعی را به زبان MATLAB بنویسید که با دریافت یک رشته از حروف کوچک انگلیسی به عنوان ورودی، آن‌را به روش Huffman کد کرده و مقدار حاصل را که یک رشته از 0 و 1 است، به عنوان خروجی بازگرداند. هم‌چنین نیاز است که تابع دیگری بنویسید که عمل دیکود کردن را برای این الگوریتم انجام دهد. طبیعتاً برای درستی‌آزمایی کد خود می‌توانید از تابع مرحله‌ی قبل استفاده کنید. دقت داشته باشید که برای پیاده‌سازی الگوریتم Huffman نیاز دارید که احتمال وقوع هر کاراکتر را داشته باشید. برای این منظور، فایلی به نام prob.mat در اختیار شما قرار داده شده است که در این فایل یک جدول به اسم prob وجود دارد که احتمال وقوع هر کاراکتر به ترتیب در آن آمده است.

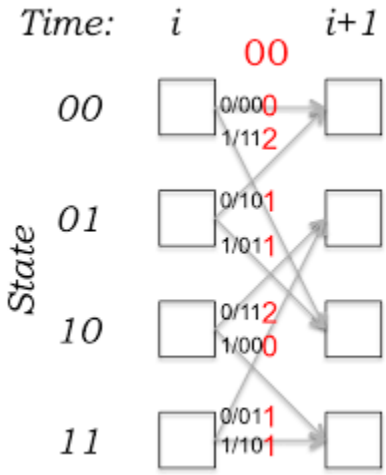
کدینگ کانال

در بخش قبل با یک روش ساده از فشرده‌سازی داده آشنا شدیم. حال در این بخش قصد داریم با یکی از روش‌های کدینگ کانال، به تشخیص خطا بپردازیم. همان‌طور که در درس آشنا شدید، کدینگ Convolutional یک روش کدینگ باحافظه است که به جای درنظر گرفتن بلاک‌های ورودی و کد کردن آن، ورودی را به شکل یک جریان داده درنظر گرفته و آن را کد می‌کند. یک از روش‌های نمایش یک انکودر Convolutional استفاده از State Machine است. برای این منظور، State Machine زیر را در نظر بگیرید:



این دیاگرام، یک شمای ساده از یک انکودر Convolutional را نشان می‌دهد. در این بخش، شما باید تابعی به زبان MATLAB بنویسید که یک رشته از 0 و1 را از ورودی دریافت کرده، آن را به کمک State Machine بالا انکود کند و جواب حاصل را به عنوان خروجی باز گرداند. دقت کنید که خروجی نیز رشته‌ای از 0 و 1 بوده و طول آن دو برابر طول ورودی می‌باشد.

در ادامه‌ی این بخش، می‌خواهیم به کمک الگوریتم Viterbi، یک رشته‌ی انکود شده را دیکود کنیم. برای پیاده‌سازی الگوریتم Viterbi از روش Dynamic Programming استفاده می‌کنیم. در این روش می‌خواهیم مقادیر Path Metric را برای هر استیت در هر زمان محاسبه کنیم. Path Metric یک مقدار است که به هر استیت نسبت می‌دهیم و برابر است با فاصله‌ی Hamming بین داده‌ی دریافتی و داده‌ای که به بیشترین احتمال در مسیر از استیت اولیه تاکنون ارسال شده است. دقت کنید که در اینجا، به هر استیت تنها از طریق دو استیت دیگر می‌توان رفت. یعنی برای محاسبه مقدار PM در یک استیت و در یک زمان مشخص، نیاز است که مقدار PM دو استیت دیگر را در زمان قبل داشت. به شکل زیر دقت کنید:



فرض کنید مقدار PM استیت s را در زمان i با PM[s, i] نشان دهیم. حال برای مثال می‌خواهیم مقدار PM[00, i+1] را محاسبه کنیم. برای این منظور نیاز است که مقادیر PM[00, i] و PM[01, i] را داشته باشیم. حال با توجه به این که در این فاصله داده‌ی 00 دریافت شده است، گذر از 00 به 00 مقدار PM را افزایش نمی‌دهد. از طرفی گذر از 01 به 00 مقدار PM را به اندازه 1 افزایش می‌دهد. درنهایت مقدار PM[00, i+1] به شکل زیر محاسبه می‌شود:

$$PM[00, i+1] = \min(PM[00, i] + 0, PM[01, i] + 1)$$

با توجه به رابطه‌ی فوق، می‌توان دریافت که مقادیر قبلی PM در محاسبه‌ی مقدار کنونی آن تاثیری ندارد و می‌توان تنها با دارا بودن مقدار آن در دقیقاً یک زمان قبل، مقدار کنونی آن را حساب کرد. این نکته می‌تواند باعث صرفه‌جویی زیادی در حافظه شود زیرا دیگر نیازی نیست که کل دیاگرام Trellis را در حافظه نگهداری کنیم. دقت داشته باشید، درصورتی که در حالتی، دو مقدار فوق برابر شدند، می‌توانید به صورت رندوم یکی را انتخاب نمایید. هم‌چنین در ابتدا برای استیت 00 مقدار PM برابر 0 و برای سایر استیت‌ها برابر بی‌نهایت است (چرا؟). با توجه به رابطه‌ی فوق، شما باید یک کد Dynamic Programming بنویسید که عمل دیکود را برای انکودری که در مرحله‌ی قبل نوشتید به

روش Viterbi انجام دهد. دقت کنید که علاوه بر محاسبه کمترین خطا، رشته‌ای که به این مقدار کمینه خطا دارد را باید به عنوان خروجی بازگردانید.

بخش پایانی

با تکمیل بخش‌های قبل، شما یک روش برای کدینگ منبع و یک روش برای کدینگ شبکه را به کمک دیکودرهای مربوطه پیاده‌سازی کرده‌اید. در این قسمت یک script به زبان MATLAB بنویسید که در آن نام خود را (با حذف space) به عنوان ورودی به انکودر منبع داده و سپس خروجی آن را به انکودر شبکه بدهید. در کنار صورت پروژه، یک فایل noise.m قرار داده شده است که باید آن را به پروژه خود اضافه کنید. این فایل از یک تابع به اسم noise تشکیل شده که یک رشته از 0 و 1 را به عنوان ورودی می‌گیرد و پس از اعمال مقدار اندکی نویز، آن را در خروجی باز می‌گرداند. شما باید خروجی مرحله قبل خود را به این تابع بدهید و مقدار حاصل را ابتدا دیکودر شبکه و پس از آن به دیکودر منبع داده و مقادیر به‌دست‌آمده در هر مرحله را در خروجی چاپ کنید. دقت داشته باشید که لزوما خروجی نهایی با نام شما یکسان نیست و می‌تواند همچنان شامل مقداری خطا باشد.

کدهای خود را در قالب فایل فشرده شده zip یا rar در اسلات مربوطه در صفحه درس آپلود کنید.