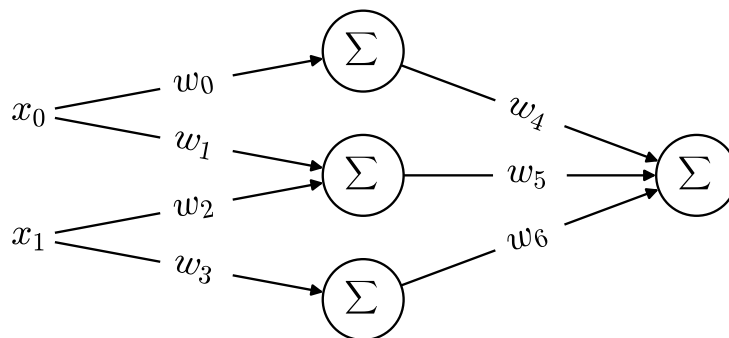




سوال اول (۱۰ نمره)

به سوالات زیر پاسخ دهید:

۱. (۵ نمره) در این سوال می‌خواهیم با استفاده از یک شبکه‌ی دولایه از نورون‌ها، تابع XOR را پیاده‌سازی کنیم. دامنه‌ی ورودی‌ها 0 و 1 می‌باشد. هر نورون در صورتی که جمع وزن‌دار ورودی‌هایش بزرگ‌تر از 0 باشد خروجی 1 تولید می‌کند و در غیر این صورت خروجی 0 است. شما باید وزن‌های w_0 تا w_6 را به گونه‌ای مقداردهی کنید که شبکه، تابع XOR را پیاده‌سازی کند. در آخر نشان دهید که شبکه به ازای تمام ورودی‌های ممکن درست رفتار می‌کند.



۲. (۵ نمره) یک دسته‌بند Logistic Regression (با ورودی برداری پیوسته و خروجی باینری)، را بر روی یک مجموعه داده آموزش داده‌ایم. سپس یک مجموعه داده‌ی جدید می‌سازیم که عین قبلی است ولی شامل یک ویژگی جدید است که برابر با نقیض برچسب است و دوباره مدل را آموزش می‌دهیم. به نظر شما عملکرد تست الگوریتم چه تغییری می‌کند (با فرض این که ویژگی جدید حاضر باشد)؟

سوال دوم (۲۰ نمره)

برای دسته‌بندی دو دسته‌ای، در فضای ورودی‌های دوبیتی، می‌توان یک دسته‌بند Naïve Bayes را با مقادیر b ، احتمال پیشین که خروجی دسته‌ی اول باشد $(Pr(y = 1))$ ، و b^{iy} ، احتمال این که یک نمونه از دسته‌ی y بیت i امش برابر با 1 باشد $(Pr(x_i = 1 | y))$ تعریف کرد.

۱. (۱۰ نمره) b ، b^{10} ، b^{11} ، b^{20} و b^{21} را به گونه‌ای مقداردهی کنید که دسته‌بند، تابع OR را پیاده‌سازی کند (00 را به دسته‌ی 0 و 01، 10 و 11 را به دسته‌ی 1 نسبت دهد).

۲. (۱۰ نمره) آیا مجموعه داده‌ی زیر، دسته‌بند درست را تولید می‌کند؟ به منظور نرم شدن، احتمال‌های 0/1 را با 0.1/0.9 جایگزین کنید.

x_1	x_2	y	x_1	x_2	y
0	0	0	1	0	1
0	1	1	1	0	1
1	0	1	0	1	1
1	1	1	0	1	1
1	0	1	0	1	1

سوال سوم (۲۰ نمره)

در این تمرین شما یک دسته‌بند حاشیه‌نرم SVM را آموزش می‌دهید که داده‌های سرطان را بر حسب نمونه‌های بافت‌برداری، به دو دسته خوش‌خیم (-1) و بدخیم (+1) دسته‌بندی می‌کند. برای این تمرین، از ماژول SVC کتابخانه sklearn استفاده کنید. مجموعه‌داده cancer-data.mat در اختیار شما قرار گرفته است. به ازای هر بیمار، نه ویژگی اندازه‌گیری شده است. برای خواندن داده‌ها از تابع scipy.io.loadmat استفاده کنید.

۱. (۸ نمره) از بین مقادیر {0.01, 0.1, 0.5, 1, 5, 10, 50} با استفاده از cross-validation مقدار مناسب C را انتخاب کنید.

۲. (۸ نمره) برای مقادیر C در قسمت قبل، نمودار خطای آموزش و تست را بر حسب C رسم کنید. تاثیر انتخاب C بزرگ روی خطای آموزش توضیح دهید. آیا این اثر با انتظار شما مطابقت دارد؟

۳. (۴ نمره) حال مجموعه آموزش و تست را عوض کنید. در واقع با استفاده از cancerInput-test دسته‌بند را آموزش دهید و با استفاده از cancerInput-train تست کنید. مشاهدات خود را گزارش کنید.

سوال چهارم (۵ + ۲۰ نمره)

در این تمرین شما با دسته‌بند MLP، ارقام دست‌نویس مجموعه‌داده‌ی MNIST را دسته‌بندی می‌کنید. برای این کار از کلاس MLPClassifier از sklearn.neural_network استفاده کنید. با استفاده از sklearn.datasets.fetch_openml('mnist_784', version=1, ...) و مقادیر پیکسل‌ها را بر ۲۵۵ تقسیم کنید تا در بازه‌ی [0, 1] قرار بگیرند. از ۶۰۰۰۰ نمونه‌ی اول برای Training و از بقیه برای Test استفاده کنید.

۱. (۲ نمره) از هر رقم موجود در مجموعه‌داده‌ی MNIST، ۱۰ تا را به طور دلخواه نمایش دهید.

۲. (۱۲ نمره) تعداد نورون‌های هر لایه را n_i^2 بگیرید و مدل را تشکیل و آن را آموزش دهید. مدل نهایی شما باید حداقل 95% داده‌های تست را به طور صحیح دسته‌بندی کنید.

۳. (۶ نمره) قبل و بعد از آموزش شبکه، تمام وزن‌های شبکه را نمایش دهید. وزن‌ها را به شکل تصاویر $n_i \times n_i$ نمایش دهید.

۴. (۵ نمره امتیازی) سعی کنید شبکه‌ای آموزش دهید که بتواند بالای 97% داده‌های تست را به طور صحیح دسته‌بندی کند. به ازای هر 0.2%، ۱ نمره‌ی امتیازی به شما تعلق می‌گیرد تا حداکثر ۵ نمره.

سوال هفتم (۳۰ نمره)

در این سوال شما یک دسته‌بند Naive Bayes برای مسئله دسته‌بندی متن، پیاده‌سازی می‌کنید. به شما یک مجموعه از مقالات متنی داده می‌شود که یا مربوط به مجله The Economist یا مربوط به مجله The Onion است. مجله اول یک مجله با زبان رسمی و جدی و مجله دوم چندان جدی نیست. هدف تمایز بین مقالات این دو مجله است.

در فضای متنی ویژگی‌ها کلمات و داده‌ها اسناد هستند که به صورت برداری از کلمات نمایش داده می‌شوند. تمام واژگانی که در این مقالات استفاده می‌شود، استخراج شده است و در اختیار شما قرار خواهد گرفت. فرض کنید تعداد این واژگان برابر با V باشد. به ازای هر مقاله، یک بردار ویژگی به صورت $X = \langle X_0, \dots, X_{V-1} \rangle$ تولید شده است. اگر کلمه i در مقاله وجود داشته باشد، X_i برابر با 1 و در غیر این صورت برابر با 0 خواهد بود. همچنین هر مقاله یک برچسب دارد که 0 نشان‌دهنده مجله The Economist و برچسب 1 نشان‌دهنده The Onion است.

مشخصاً در این مسئله ما توزیع توام دقیق ویژگی‌ها و برچسب‌ها را نداریم. بنابراین باید پارامترهای $P(X|Y=y)$ و $P(Y=y)$ را از روی داده‌های آموزش تخمین بزنیم. برای هر اندیس $w \in \{1, \dots, V\}$ و $y \in \{1, 2\}$ ، توزیع X_w با شرط $Y=y$ یک توزیع برنولی با پارامتر θ_{yw} است. به بیان دیگر:

$$P(X_w = 0|Y = y) = 1 - \theta_{yw} \text{ و } P(X_w = 1|Y = y) = \theta_{yw}$$

معتقدیم که هر کلمه با احتمال ناصفر (اما شاید بسیار کوچک) می‌تواند در یک مقاله از هر کدام از دو مجله ظاهر شود. برای اطمینان از این که تخمین‌هایی که از پارامترها می‌زنیم همیشه ناصفر هستند، یک توزیع پیشین $Beta(2, 1)$ برای θ_{yw} متصور هستیم و از داده‌های آموزش تخمین MAP خواهیم زد. به طور مشابه توزیع Y یک توزیع برنولی با پارامتر ρ است. به بیان دیگر:

$$P(Y = 0) = 1 - \rho \text{ و } P(Y = 1) = \rho$$

در این مورد از آنجایی که تعداد زیادی مقاله از هر دو مجله داریم، از تخمین MLE استفاده می‌کنیم. برای این تمرین یک پوشه Data در اختیار شما قرار خواهد گرفت که شامل موارد زیر است:

- **Vocabulary.txt**: یک فایل متنی شامل کلماتی که در مجلات خبری ظاهر می‌شوند. هر خط از این فایل یک کلمه است. توجه کنید که روی کلمات یک الگوریتم ریشه‌یابی اعمال شده است و عجیب بودن برخی کلمات به این علت است. مثلاً کلمه Stemming و Stemmed هر دو به Stem تغییر می‌کنند. شماره خط id هر کلمه است که عددی بین 0 تا 26047 خواهد بود.

- **XTrain**: یک ماتریس $n \times V$ که همه n سندی را نشان می‌دهد که برای آموزش مدل از آن استفاده خواهید کرد. دقت کنید که درایه (i, j) این ماتریس برابر با 1 است اگر کلمه j در سند i ظاهر شود و در غیر این صورت برابر با صفر خواهد بود.

- **yTrain**: یک ماتریس $n \times 1$ که شامل برچسب‌های سندهاست. $yTrain(i, 1) = 0$ اگر سند i مربوط به مجله The Economist باشد. در غیر این صورت $yTrain(i, 1) = 1$.

- **XTest** و **yTest** مشابه XTrain و yTrain است. تنها تفاوت آن است که این ماتریس‌ها هر یک m سطر دارند. این داده‌ها برای تست کردن مدل شماست و نباید از آن‌ها برای آموزش مدل استفاده کنید.

برای لود کردن XTrain و XTest از تابع loadmat داده‌شده و برای لود کردن ماتریس‌های برچسب از تابع np.load استفاده کنید.

محاسبات در فضای لگاریتم

هنگام کار با اعداد خیلی بزرگ یا خیلی کوچک (مانند احتمالات)، بهتر است در فضای لگاریتم کار کنیم تا دچار مسائل مربوط به دقت اعداد نشویم. در فضای لگاریتمی به جای خود اعداد از لگاریتم آن‌ها استفاده می‌کنیم. مثلاً اگر $p(x)$ و $p(y)$ مقادیر احتمال باشند، به جای ذخیره این دو و محاسبه $p(x)p(y)$ ، در فضای لگاریتمی کار می‌کنیم و $\log(p(x))$ و $\log(p(y))$ را ذخیره می‌کنیم و لگاریتم حاصل ضرب را محاسبه می‌کنیم: $\log(p(x) * p(y)) = \log(p(x)) + \log(p(y))$

۱. (۲ نمره) با توجه به این که تعداد واژگان V است، تفاوت در تعداد پارامترهای تخمین زده شده را با فرض Naive بودن ویژگی‌ها و بدون آن بیان کنید.

۲. (۵ نمره) تابع $\text{NB_XGivenY}(\text{XTrain}, \text{yTrain})$ را کامل کنید. خروجی این تابع D یک ماتریس $2 \times V$ است که برای هر کلمه $w \in \{0, \dots, V-1\}$ و برچسب $y \in \{0, 1\}$ درایه $D(y, w)$ تخمین MAP برای $\theta_{y,w} = \log[P(X_w = 1 | Y = y)]$ با توزیع پیشین $\text{Beta}(2, 1)$ است. راهنمایی: با توجه به توزیع پیشین و توزیع برنولی برای Likelihood، تخمین MAP برای توزیع پسین مد نظر به صورت زیر خواهد بود. در این رابطه $N_{y,w}$ تعداد دفعاتی است که کلمه w در سندی از دسته y ظاهر می‌شود. همچنین N_y تعداد کل اسناد متعلق به دسته y است.

$$\hat{\theta}_{y,w}^{\text{MAP}} = \frac{N_{y,w} + 2 - 1}{N_y + (2 - 1) + (1 - 1)}$$

۳. (۳ نمره) تابع $\text{NB_YPrior}(\text{yTrain})$ را تکمیل کنید. خروجی این تابع p تخمین MLE برای $\log[\rho] = \log[P(Y = 1)]$ است. راه سرراستی برای این کار آن است که تخمین MLE برای $P(Y = 1)$ را محاسبه کنید و سپس لگاریتم آن را خروجی دهید.

۴. (۶ نمره) تابع $\text{NB_Classify}(\text{X}, \text{D}, \text{logp})$ را تکمیل کنید. ورودی X این تابع یک ماتریس $n \times V$ شامل n بردار داده است. خروجی آن یک بردار $n \times 1$ از برچسب‌های پیش‌بینی شده برای داده‌های ورودی است.

۵. (۲ نمره) تابع $\text{ClassificationError}(\text{yHat}, \text{yTruth})$ را تکمیل کنید که دو بردار با اندازه یکسان ورودی می‌گیرد و مشخص می‌کند چند درصد از درایه‌های دو بردار برابرند.

۶. (۸ نمره) با اجرای توابعی که تکمیلی کردید، مدل خود را آموزش دهید:

```
D = NB_XGivenY(XTrain, yTrain);  
logp = NB_YPrior(yTrain);
```

از مدل برای دسته‌بندی بردار ویژگی مقالات داده شده در XTrain و XTest استفاده کنید:

```
yHatTrain = NB_Classify(XTrain, D, logp);  
yHatTest = NB_Classify(XTest, D, logp);
```

از تابع $\text{ClassificationError}$ استفاده کنید تا خطای آموزش و تست را محاسبه کنید.

```
trainError = ClassificationError(yHatTrain, yTrain);  
testError = ClassificationError(yHatTest, yTest);
```

میزان خطای تست و خطای آموزش را با یکدیگر مقایسه کنید. دلیل تفاوت این دو را توضیح دهید.

۷. (۴ نمره) قسمت قبل را این بار با آموزش مدل روی داده‌های `XTrainSmall` و `yTrainSmall` تکرار کنید. دلیل تفاوت خطای آموزش و تست در این حالت و حالت قبل را شرح دهید.

موفق باشید :)