به نام خدا

تمرین سری اول درس یادگیری عمیق

دکتر محمدی

محمد یارمقدم

96462104

سوال یک)

1) در دهه 1960 میلادی و اوایل دهه 1970 میلادی، متخصصین زیادی علم هوش منصوعی را نزدیک به انتها دانستند. زیرا پس از پروژه ها و پیشرفت های فراوانی این حوزه، دوره سکون و شکست پروژه ها فرا رسیده بود. هم چنین بسیاری از انتظار های به نتیجه نرسیده بود. در این هنگام این علم به اولین دوره رکود یا اصطلاحا AI WINTER رسید.
اما این تنها رکود این علم نبود. در دهه 1980 یک جهش جدید در هوش مصنوعی نمادین باعث شد که بخار در میان کارخانه های بزرگ وارد شود. در ابتدا موفقیت های ابتدایی مشاهده شد و این امر باعث شد تحقیق و مطالعه در سراسر دنیا آغاز شود. در 1985 کارخانه ها حدود 1 میلیارد دلار در حوزه تکنولوژی در سال خرج می کردند اما در 1990 اثبات شد که هزینه نگهداری این سیستم ها بالاست و در حوزه محدود قابلیت استفاده دارند و علاقه و موج به وجود آمده فروکش کرد. در نتیجه دوره دوم رکود آغاز شد. در حال حاضر نیز سومین دوره هیجان اضافه در هوش مصنوعی احساس میشود و ما هنوز در مرحله خوش بینی شدید هستیم. بهتر است انتظاراتمان را برای کوتاه مدت کنترل کنیم و کاهش دهیم و اطمینان حاصل کنیم که مردم کمتر با جنبه های فنی آشنا میشوند.

2) راه اصلی برای بهبود نتیجه خروجی شبکه عصبی این است که مقدار loss را به عنوان یک سیگنال فیدبک برای تنظیم وزن های لایه های شبکه به آن وارد کرد. در جهتی که loss کمتر شود. این تنظیم بر عهده optimizer است.
الگوریتم backpropagation توسط ان پیاده سازی میشود. این الگوریتم محوریت و مرکزیت الگوریتم های یادگیری عمیق را بر عهده دارد. این الگوریتم در اواسط دهه 80 میلادی ابداع شد. در واقع این الگوریتم روشی برای آموزش زنجیره عملیات پارامتریک با استفاده از gradient descent. از این تاریخ به شبکه های عصبی وارد شد.
این الگوریتم با محاسبه مقدار loss نهایی آغاز می کند و به صورت بازگشتی از لایه های بالایی تا لایه های پایینی کار می کند. قاعده زنجیره ای را برای محاسبه سهم هر پارامتر اعمال می کند.

3) برای کنترل خروجی یک شبکه عصبی، نیاز داریم که تفاوت مقدار اندازه گیری و پیش بینی شده را از مقدار قابل انتظار بدست آوریم. این وظیفه بر عهده "loss function" است که با آن "objective function" نیز می گویند. این تابع در ورودی پیش بینی و مقادیر تولیدی شبکه عصبی را دریفات و یک مقدار فاصله را تولید می کند که در واقع معیاری از اینکه شبکه تا چه حد خوب عمل کرده است.

4) توابع مرکزی یا "kernel methods" گروهی از الگوریتم های دسته بندی هستند که معروف ترین آنها "SVM" است. این الگوریتم بر روی دسته بندی اطلاعات از روی پیدا کردن محدوده های تصمیم گیری یا "decision boundaries" متمرکز است.
این نواحی می توانند یک خط یا یک سطح باشند که دو فضا را از هم جدا می کنند. این تکنیک از مپ کردن دیتا برای یک مسئله دسته بندی با ابعاد بالا شاید در کاغذ خوب باشد اما در واقع از نظر محاسبات و عملیاتی رام نشدنی هستند. در اینجا نیازی به محاسبه مختصات جدید در فضای جدید نیست. فقط محاسبه فاصله نقاط در فضای جدید لازم است که به راحتی توسط kernel function انجام می شود. این تابع یک عملیات محاسبه پذیر است که هر دو نقطه از فضای اولیه را به فاصله آن دو در فضای جدید تبدیل می کند.

5) در اصل، Tensor یک کانتینر از اطلاعات است که همیشه اطلاعات عددی است. پس یک کانتینر از اعداد است. ماتریس های دو بعدی در واقع Tensor هستند. Vector ها در واقع Tensor های یک بعدی هستند.

Vector که دارای چهار entry باشد "4 Dimensional vector" گویند. این وکتور فقط یک محور دارد و حول این محور 4 بعد دارد. اما Tensor چهار بعدی در واقع 4 محور دارد و می تواند هر تعدادی بعد حول هر کدام از این محور ها داشته باشد.
برای Tensor چهار بعدی اگر هر خانه یک ماتریس را یک آرایه در نظر بگیریم، یک Tensor سه بعدی را تولید کردیم. حال اگر هر خانه یک آرایه را یک Tensor سه بعدی قرار دهیم، Tensor چهار بعدی را خواهیم داشت.

6) عملیات dot که به آن "tensor product" هم می گویند، پرکاربرد ترین و رایج ترین عملیات tensor است. یک "element-wise product" با * نمایش داده میشود و دو بردار را نظیر به نظیر ضرب داخلی می کند. اما Tensor product یک فضای بردار است که می توان آن را فضای همه تنسورها دانست که می توانند از بردارهای فضاهای تشکیل دهنده آن با استفاده از یک عملیات اضافی ساخته شوند که می تواند به عنوان تعمیم و انتزاع محصول خارجی در نظر گرفته شود.

سوال سه)

Data Type :

که به آن dtype نیز می گویند. در واقع نوع داده ای است که داخل Tensor موجود است. می تواند مقادیری مثل float32، uint8، float64 و حتی char داشته باشد.

Rank :

تعداد محور های یک Tensor را Rank گویند. به این مقدار Tesnor's ndim نیز می گویند. برای ماتریس این مقدار 2 و برای تنسور سه بعدی این مقدار 3 است.

Shape :

یک tuple از اعداد صحیح است که مشخص می کند اطراف هر محور Tensor چند بعد موجود است.

سوال چهار)

در باکس اول، لود دیتاست انجام می پذیرد. این کار بوسیله کتابخانه sklearn صورت گرفت. سپس 4 مورد اول هر سطر این دیتاست که مقادیر عددی برای متغیر های هر مورد است را جدا کردیم تا آموزش مدل روی آن انجام شود.

در باکس دوم کلاس مدل Naïve Bases نوشته شده است. این کلاس شامل توابع زیر است:

```
1) separate_classes
2) stat_info
3) fit
4) distribution
5) predict
6) accuracy
```

در ادامه هر تابع در خلال مراحل توضیح داده میشود.

در قسمت main که باکس آخر محسوب میشود، در ابتدا داده تست و آموزش از هم جدا میشوند. به این صورت که 80 درصد داده متعلق به آموزش و مابقی برای تست درنظر گرفته شد. سپس X و Y در تست و آموزش از یکدیگر جدا میشوند. سپس یک نمونه از کلاس ساخته و در اولین مرحله تابع fit صدا زده میشود. این تابع X و Y آموزش را دریافت میکند.

سپس در این تابع، seprate_classes فراخوانی میشود که به ازای هر Y غیر تکراری یک تاپل را ساخته و تمامی مقادیر X متناظر با آن را در value آن ذخیره میکند و در نهایت دیکشنری آن را بر میگرداند.

سپس در ادامه تابع فیت در یک دیکشنری دیگر به نام class_summary به ازای هر کدام از اعضای این دیکشنری prior_prob محاسبه میشود که درصد این متغیر از نظر تکرار در کل X است.

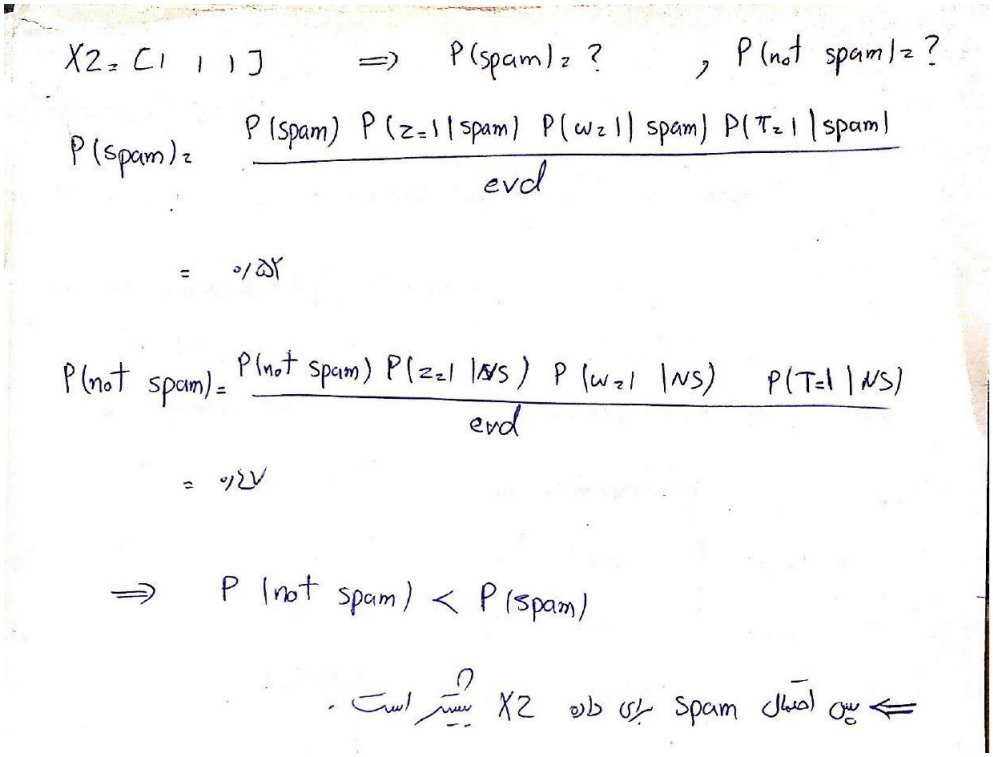
پس از این تابع، تابع predict فراخوانی میشود. در این تابع به ازای تمامی item های class_summary ، همسایگی را برابر 1 و یک متغیر برای طول متغیر های هر سطر در نظر گرفته میشود.

سپس با توجه به توضیحات درس، تابع distribution با دو مقدار mean و std برای هر ایندکس فراخوانی میشود و طبق فرمول زیر احتمال محاسبه میشود و همسایگی(likelihood) را آپدیت میکند.

در نهایت از روی تعداد برابری y_te

st و y_pred دقت مدل محاسبه میشود.

سوال دو)



ادامه سوال دو)

$X2 = [1\ 1\ 1] \Rightarrow P(spam)_2 ?\ ,\ P(not\ spam)_2 ?$

$$P(spam)_2 = \frac{P(spam)\ P(z=1|spam)\ P(w_2=1|spam)\ P(T_2=1|spam)}{evd}$$

$= 0.052$

$$P(not\ spam)_2 = \frac{P(not\ spam)\ P(z_2=1|NS)\ P(w_2=1|NS)\ P(T=1|NS)}{evd}$$

$= 0.127$

$\Rightarrow P(not\ spam) < P(spam)$

$\Rightarrow$ پس احتمال spam برای داده X2 بیشتر است.