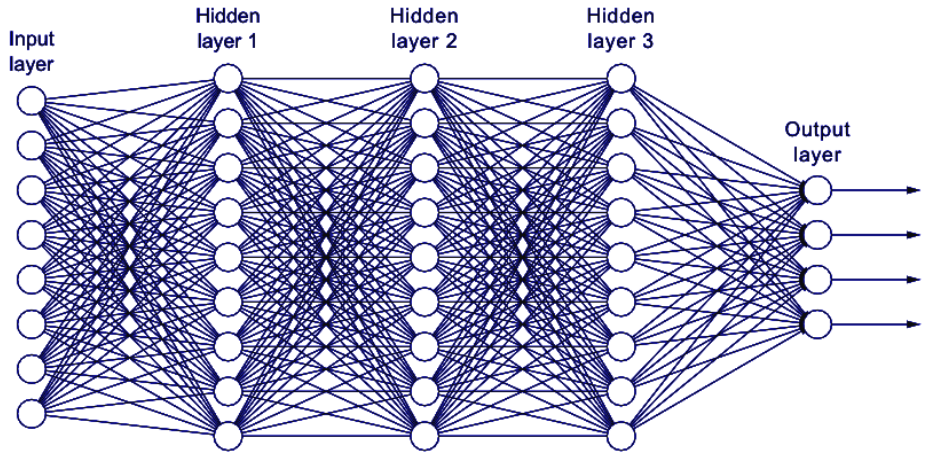
مدرس: **دکتر فدایی**
طراح: حسام اسداله‌زاده، ادیب رضایی، مسعود طهماسبی‌فرد، نوید دهبان

مهلت تحویل: شنبه ۶ خرداد ۱۴۰۲، ساعت ۲۳:۵۹

مقدمه

شبکه عصبی Feed Forward:

شبکه‌های عصبی یکی از قدرتمندترین ساختارهای یادگیری ماشین هستند که در سال‌های اخیر با توجه به افزایش قدرت محاسباتی پردازنده‌ها، کاربردهای بسیاری در حوزه‌های مختلف علمی و صنعتی پیدا کرده‌اند. از مزایای این الگوریتم‌ها آن است که امکان ساخت هر تابع مشتق‌پذیر دلخواهی را با تنها استفاده از دو لایه‌ی مخفی از نورون‌ها ممکن می‌سازد[1]. بنابراین امکان پیاده‌سازی و یادگیری توابع و طبقه‌بندهای غیرخطی با کمک آن‌ها ممکن است. در پروژه پنجم به پیاده‌سازی شبکه‌های عصبی Feed Forward می‌پردازیم. در بخش اول این پروژه، با استفاده از یک شبکه‌ی عصبی که به صورت دستی پیاده‌سازی خواهید کرد، به طبقه‌بندی نقاط دو کلاس با توزیع داده‌های هلالی‌شکل و سپس در بخش دوم به طبقه‌بندی متون با استفاده از کتابخانه nltk و torch خواهید پرداخت.



برای آموزش یک شبکه‌ی عصبی، باید مقادیر و ویژگی‌های **عددی** به ورودی شبکه اعمال شود. بنابراین ابتدا باید ویژگی‌های موجود در متون (نظرات کاربران در سایت IMDB) استخراج شده و سپس به عنوان ورودی به شبکه داده شوند. برای استخراج ویژگی[2] از متون، روش‌های مختلفی موجود است. شبکه قرار است بر اساس این ویژگی‌ها و با ساختن ترکیبات غیرخطی از آن‌ها، وزن اتصالات بین لایه‌ها و پارامترهایش را طوری تنظیم کند، که خروجی آن ضمن داشتن کمترین خطا، کلاس متون ورودی متناظر را به درستی پیش‌بینی کند.

تعریف مسئله

در این تمرین، در بخش **اول** به پیاده‌سازی یک شبکه‌ی عصبی Feed Forward از پایه و با استفاده از کتابخانه NumPy می‌پردازید. جهت تسریع این فرایند، یک Notebook ناقص از پیاده‌سازی شبکه نیز در اختیار شما قرار می‌گیرد که لازم است از آن استفاده نمایید. در بخش **دوم**، به کمک کدهای بخش **اول**، یک شبکه‌ی عصبی را روی داده‌های هلالی‌شکل آموزش خواهید داد. در بخش **آخر** نیز با کتابخانه‌های آماده برای پیاده‌سازی شبکه‌های عصبی آشنا خواهید شد و تاثیر برخی از عوامل را در فرایند یادگیری بررسی خواهید کرد. برای راحتی استفاده از کتابخانه‌ها و انجام محاسبات، می‌توانید از سرویس **Google Colab** استفاده کنید.

---

[1] Universal Function Approximator

[2] Feature Extraction

معرفی مجموعه داده

در این تمرین، در بخش اول شما به حل یک مسئله طبقه‌بندی دو کلاسه با مرز تصمیم غیرخطی خواهید پرداخت. کدهای لازم برای تولید دیتاست مورد استفاده در این بخش، در نوت‌بوکی که در اختیار شما قرار گرفته موجود است. همچنین توابعی جهت ترسیم توزیع داده و مرز تصمیم به دست آمده توسط شبکه‌ی آموزش داده شده در نوت‌بوک وجود دارد.

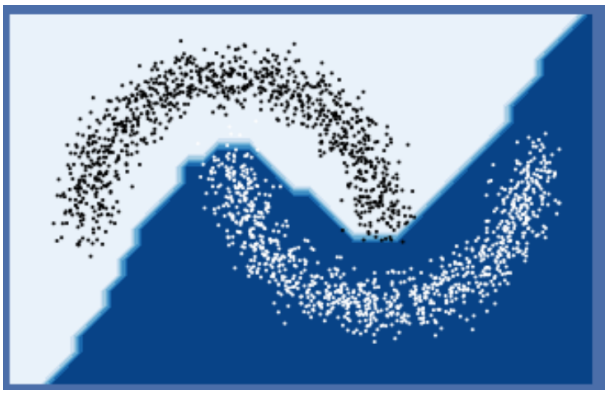
در بخش دوم شما با مجموعه داده نظرهای IMDB کار خواهید کرد. مجموعه داده شامل تقریبا **۵۰ هزار** نظر کاربران درباره فیلم‌های IMDB می‌باشد. این مجموعه داده دارای دو ستون review و sentiment می‌باشد. ستون review حاوی نظر کاربر و ستون sentiment نشان دهنده مثبت یا منفی بودن نظر است. اطلاعات بیشتر راجع به مجموعه داده را میتوانید اینجا بخوانید.

به منظور سهولت کار، مجموعه داده به صورت یک فایل csv به نام `imdb_dataset.csv` در اختیارتان قرار گرفته است.

بخش اول: تکمیل بخش‌های ناقص شبکه عصبی

یک فایل Notebook شامل کدهای ناقص مورد نیاز برای پیاده سازی شبکه عصبی Feed Forward آپلود شده و در این قسمت با تکمیل بخش‌های مختلف این فایل، در نهایت یک کلاس `Sequential` خواهید داشت که به کمک آن می‌توانید شبکه‌های عصبی Feed Forward پیاده کنید و آموزش دهید. پارامترهای شبکه موردنظر از طریق روش SGD[3] در طی فرایند آموزش به‌روزرسانی خواهند شد. بخش‌های حذف شده از کد که لازم است آن‌ها را کامل کنید، با `# YOUR CODE HERE` مشخص شده‌اند.

پس از پیاده‌سازی و تکمیل بخش‌های ناقص، باید نمودار تغییرات loss را رسم کرده و مرز تصمیم به دست آمده توسط شبکه‌ی آموزش داده شده را نمایش دهید.



شکل ۱. خروجی نمونه مورد انتظار در بخش اول

[3] Stochastic Gradient Descent

بخش دوم: بررسی و پیش‌پردازش داده متنی

در بخش دوم باید اطلاعات متنی داخل مجموعه داده را برای تحلیل‌های بعدی پیش‌پردازش کنیم. برای این کار می‌توانید از روش‌های مختلفی استفاده کنید. روش TF-IDF (Term Frequency-Inverse Document Frequency) یکی از روش‌های استخراج ویژگی و وزن‌دهی به کلمات در متن‌ها است که در حوزه پردازش زبان طبیعی و هوش مصنوعی استفاده می‌شود. با استفاده از قطعه کد زیر می‌توانید از این روش برای پیش‌پردازش متون استفاده کنید:

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

❓ **به طور مختصر در مورد نحوه عملکرد TF-IDF توضیح دهید.**

⬅ توجه: دقت کنید که به صورت پیش‌فرض TfidfVectorizer کل کلمات موجود در داده‌ها را به عنوان ویژگی در نظر می‌گیرد و به ازای جملات مختلف و تعداد تکرار کلمات در هر جمله، کلمات را وزن‌دهی می‌کند. اما تعداد زیادی از کلمات بسیار کم‌تکرار هستند و اهمیت کم‌تری در آموزش مدل دارند. در نتیجه، برای کاهش مصرف حافظه و محاسبات، می‌توانید آرگومان `max_features` را برابر مقداری بین 3000 تا 7000 قرار دهید.

شما باید نظرات را تا حد ممکن Normalize کنید. روش‌های ممکن، شامل حذف کلمات پرتکرار یا همان ایست‌واژه‌ها[4]، حذف عبارات بی‌معنی مثل </ br> که در نظرات وجود دارد، حذف کاراکترهای بی‌اهمیت مانند \n و \r، تبدیل کلمات به ریشه‌ی آن‌ها و ... است. برای این کار می‌توانید از کتابخانه NLTK (Natural Language Toolkit) که یکی از کتابخانه‌های معروف و قدرتمند در زمینه پردازش داده متنی و زبان طبیعی است، استفاده کنید. این کتابخانه شامل مجموعه‌ای از ابزارها و منابع برای تحلیل، پردازش و استخراج اطلاعات از متون زبان طبیعی است.

همچنین داده‌های ستون sentiment باید به داده‌های عددی تبدیل شوند. برای این کار می‌توانید از **label encoding** استفاده نمایید.

دقت کنید که مراحل پیش‌پردازش هم روی داده‌های train و هم روی داده‌های test باید انجام شود و لزوماً اجرای هر نوع پیش‌پردازشی باعث بالا رفتن دقت مدل شما نخواهد شد. روش‌های متفاوت را با استفاده از کتابخانه یا بدون آن امتحان کنید و ترکیب هر کدام از آن‌ها که به مدل شما بیشتر کمک می‌کرد را اجرا کنید.

❓ در گزارش کار خود، جایگزین کردن کلمات با روش Stemming و Lemmatization را توضیح دهید. یکی از این روش‌ها را به اختیار انتخاب کرده و به عنوان آرگومان tokenizer به TfidfVectorizer پاس دهید.

- یک نظر مثبت و یک نظر منفی رندوم را در مجموعه داده train بررسی کنید و نشان دهید.
- تعداد نظرات هر دسته را برای مجموعه داده train و test محاسبه کنید و برای آن‌ها نمودار میله‌ای رسم کنید.

❓ بررسی کنید که قبل از دادن ورودی به شبکه عصبی، مقدار هر feature بین 0 تا 1 باشد. **در صورت برقرار نبودن این شرط چه مشکلی ممکن است رخ دهد؟**

[4] stop words

بخش سوم: استفاده از کتابخانه PyTorch

کتابخانه Torch

با توجه به پیشرفت‌های اخیر هوش مصنوعی و کاربرد روز افزون آن در صنعت، frameworkهای بسیار قدرتمندی برای سهولت در ساخت و آموزش شبکه‌های عصبی بسیار پیچیده، با کارایی بسیار بالا، عرضه شده است. در این بخش قصد داریم با کتابخانه PyTorch برخی از مسائل مربوط به شبکه‌های عصبی را بررسی نماییم.

در این بخش باید به کمک torch یک شبکه Feed Forward ایجاد کنید که **حداقل شامل دو لایه مخفی** باشد (با در نظر گرفتن لایه‌های ورودی و خروجی (sigmoid) باید شبکه شما حداقل دارای ۴ لایه باشد. همچنین از تابع فعال‌ساز ReLU در تمام لایه‌های مخفی استفاده شود).

دقت کنید به هیچ عنوان نباید از لایه‌های شبکه‌های Convolution و Recurrent استفاده کنید.

قسمت اول) آموزش شبکه

- یک شبکه‌ی عصبی با شرط‌های گفته شده را طراحی کنید و آموزش دهید.
- سعی کنید معماری شبکه (تعداد لایه‌ها و تعداد نورون‌ها در هر لایه) را طوری تغییر دهید که به دقت بهتری برسید.
- نتایج Recall، Precision، Accuracy و F-1 را روی داده آموزش و تست گزارش کنید.

قسمت دوم) وزن‌دهی شبکه

مقدار اولیه وزن‌ها در آموزش شبکه اهمیت دارد.

- اگر مقدار اولیه تمام وزن‌های شبکه برابر صفر بود و شبکه را آموزش می‌دادید، چه نتیجه‌ای حاصل می‌شد؟ **(نیازی به پیاده‌سازی نیست)**

قسمت سوم) تاثیر learning rate

یکی از پارامترهای مهم در آموزش دادن شبکه‌های عصبی، learning rate می‌باشد.

- رفتار شبکه را برای learning rate با مقدار بالاتر (مثلا 10 برابر) و پایین‌تر (مثلا 0.1 برابر) نسبت به حالت قبل را بدست آورید. نتیجه خود را با حالت قبل مقایسه کنید و توجیه کنید.

برای تمام قسمت‌های بعد، از learning rate بهینه‌ای که بدست آورده‌اید استفاده کنید.

قسمت چهارم) تاثیر activation function

- عملکرد شبکه‌ی طراحی‌شده در قسمت قبل را به کمک Activation Function های زیر بسنجید و نتایج را مقایسه نمایید.
  - تابع فعال‌ساز Sigmoid
  - تابع فعال‌ساز Hyperbolic Tangent
  - تابع فعال‌ساز Leaky ReLU
- چرا توابع Tanh و Sigmoid به طور کلی، انتخاب مناسبی برای استفاده در لایه‌های مخفی نیستند؟
- تفاوت و برتری Leaky ReLU نسبت به ReLU چیست؟

قسمت پنجم) تاثیر batch size

- عملکرد شبکه را به ازای batch size با مقادیر 16 و 256 بدست آورید. نتیجه خود را با حالت قبل مقایسه کنید و توجیه کنید.
- علت استفاده از batch در فرایند آموزش چیست؟ مزایا و معایب batch size بسیار کوچک و بسیار بزرگ را شرح دهید.

منابع

برای یادگیری شروع کار با Torch میتوانید از این لینک کمک بگیرید.

برای آموزش شیوه استفاده از Google Colab می‌توانید از این لینک یا این لینک استفاده نمایید.

نکات پایانی

- دقت کنید که هدف پروژه **تحلیل نتایج** و تاثیر عوامل مختلف است؛ بنابراین از ابزارهای تحلیل داده بطور مثال نمودارها استفاده کنید و توضیحات مربوط به هر بخش از پروژه را بطور خلاصه و در عین حال مفید در گزارش خود ذکر کنید.
- نتایج و گزارش خود را در یک فایل فشرده با عنوان AI-CA5-P1-<#SID>.zip تحویل دهید. محتویات پوشه باید شامل فایل jupyter-notebook، خروجی html و فایل‌های مورد نیاز برای اجرای آن باشد. **تحلیل و نمایش خروجی‌های خواسته شده بخشی از نمره این تمرین را تشکیل می‌دهد.** از نمایش درست خروجی‌های مورد نیاز در فایل html مطمئن شوید.
- توجه داشته باشید که علاوه بر ارسال فایل‌های پروژه، این پروژه به صورت حضوری نیز تحویل گرفته خواهد شد. بنابراین تمام بخش‌های پروژه باید قابلیت اجرای مجدد در زمان تحویل حضوری را داشته باشند. همچنین در صورت عدم حضور در تحویل حضوری نمره‌ای دریافت نخواهید کرد.
- در صورتی که سوالی در مورد پروژه داشتید بهتر است در فروم یا گروه درس مطرح کنید تا بقیه از آن استفاده کنند؛ در غیر این صورت با طراحان در ارتباط باشید.
- هدف از تمرین، یادگیری شماست. لطفا تمرین را خودتان انجام دهید.

موفق باشید.