

# به نام خدا



درس: هوش مصنوعی

نام:

محمد مهدی شامخی

دانشکده مهندسی کامپیوتر

پاییز 1402

## 1. بخش اول:

در این قسمت با استفاده از شبکه عصبی دیتاست هواپیما را تست کردیم. همچنین برای پیاده سازی شبکه عصبی از کتابخانه ی Sklearn استفاده کرده ایم. مانند الگوریتم درخت تصمیم پیاده سازی شده برای این سوال تقریباً ۰.۱ داده برای تست مدل و مابقی برای آموزش به شبکه داده شده است. دقتی که شبکه به دست آورده است تقریباً ۹۶ درصد است. در حالی که دقت درخت تصمیم پیاده سازی شده برای این دیتاست تقریباً ۹۳ درصد بود. این درحالی است که کل فرایند آموزش و تست در ۱ دقیقه برای مدل شبکه عصبی طول کشید و برای درخت تصمیم در ۷ دقیقه!

در مدل شبکه عصبی نسبت به درخت تصمیم، ۳ درصد پیشروی داشتیم. شبکه عصبی پیاده سازی شده دو لایه پنهان با تعداد نوروں های ۶۴ و ۱۲ دارد. همچنین تابع فعال سازی استفاده شده برای تمام نوروں ها relu است. همچنین تعداد حداکثر مشاهده نیز ۱۰۰۰ است. دلیل اینکه مدل شبکه عصبی طراحی شده نسبت به درخت تصمیم بهتر عمل کرده این است که این مدل متغیر های وزن خود را طوری انتخاب میکند که تابع خطا مینیمم شود و از طرفی تعداد ایتريشن هایی که برای آموزش دیتا انجام می دهد نیز بیشتر است. از لحاظ ریاضیاتی محاسبات قوی تری نسبت به درخت تصمیم انجام می دهد. همچنین در شبکه عصبی تعداد نوروں ها و لایه ها بسیار تاثیر گذار است در خروجی. تا حدی که بتوان دقیقاً خروجی را پیش بینی کرد. البته درخت تصمیم نیز در این نوع مسایل classification خوب عمل میکنند. استفاده از آنتروپی برای انتخاب فیچر در هر نود درخت تا متد هرس درخت. در حالت کلی هر دو آنها به نوعی خوب هستند اما شبکه عصبی به دلیل دست باز بودن در تعداد نوروں ها و لایه ها مدل قوی تری خواهد بود.

## 2. بخش دوم:

در این قسمت با استفاده از شبکه عصبی رفتار توابع  $\sin(2x+5)$  و  $x^2$  و  $e^x$  را پیاده سازی کردیم. به طور مثال برای تابع  $\sin$  گفته شده، به صورت رندوم ۱۰۰ نقطه بین بازه ۰ تا ۱۰ روی این تابع تولید کرده و به عنوان داده آموزشی به شبکه داده شده است. همچنین ۱۰۰ نقطه به طور تصادفی در همین بازه تولید کرده و به عنوان داده تست به شبکه داده ایم. MSE تولید شده برای این مدل ۰.۰۱ چیزی نزدیک به صفر است. همچنین با استفاده از کتابخانه matplotlib خروجی را در Predict\_Sin2 ذخیره کردیم و تقریباً می توان گفت که دیتا پیش بینی شده با دیتا تست یکسان است. البته دقت کنید که این بهترین پیش بینی به دست آمده بوده است. بعضاً داشته ایم نتیجه ای با MSE بالا تر که خوب نبوده.

هر چه تعداد نقاط ورودی شبکه را بیشتر کرده، از آن طرف نیز به نتیجه بهتری می رسیم. شکل TedadNoghat2.png این را به وضوح نشان می دهد. تفاوتی که اینجا داشته ایم این بوده است که تعداد نقاط آموزش را ۱۰۰ برابر کرده و به ۱۰۰۰۰ رساندیم. همچنین MSE در این حالت به ۰.۰۰۱ رسیده است. هر چه تعداد لایه ها و تعداد نوروں ها را بیشتر کنیم می تواند به بهبودی شبکه کمک کند. شکل x2\_predict2.png به ما مدل شبکه عصبی خوبی از تخمین تابع  $x^2$  با لایه های پنهان ۱۱۰، ۱۱۰، ۱۵۰، ۲۰۰ نشان می دهد. همچنین در این حالت MSE برابر ۰.۰۱ هست. همچنین شکل x2\_bad\_predict2.png به ما مدل شبکه عصبی بدی از تابع  $x^2$  با لایه های پنهان ۱۰، ۱۰ نشان می دهد. همانطور که میبینید تعداد نوروں ها و لایه های کم نسبت به حالت قبل باعث شده تابع خطی پیش بینی شود. همچنین در این حالت MSE برابر ۴۷ هست! هر چه تعداد تکرار ها برای آموزش شبکه بیشتر باشد، دقت شبکه بالاتر خواهد بود. شکل ex\_predict2.png به ما مدل شبکه عصبی خوبی از تخمین تابع  $e^x$  با حداکثر تکرار ۱۰۰۰ را می دهد. MSE در این مدل ۰.۰۳ است. با اینکه شکل

ex\_bad\_predict2.png مدل شبکه عصبی ضعیف تری با حداکثر تکرار ۱۰۰ نسبت به حالت قبل برای تابع  $e^x$  نشان می دهد. همچنین در این حالت MSE ۳۲ است. هر چه قدر وسعت دامنه ورودی بیشتر باشد، دقت کمتر خواهد بود به خصوص برای توابع پیچیده. شکل vosatziad2.png این موضوع را برای تابع  $e^x$  در بازه ی  $x$  بین -۵۰ تا ۵۰ نشان می دهد. دلیل این امر این است که مقدار به صورت نمایی زیاد شده و باعث MSE بسیار بالا می شود. در این مدل MSE یک عدد ۴۰ رقمی است. عوامل بسیار دیگری نیز وجود دارد که در دقت شبکه تاثیر گذار باشد مانند تابع فعال سازی و .....

### 3. بخش سوم:

در این قسمت تاثیر نویز بر شبکه عصبی را برسم کردیم. به طور وضوح اگر نویز در دیتا ترین وجود داشته باشد، کار شبکه سخت تر است، اما به خوبی می تواند آن را تشخیص دهد. شکل Noise\_NoghatZiad3.png مدل شبکه عصبی آموزش داده شده روی ۱۰۰۰۰ داده به همراه ۵ لایه که هر لایه شامل ۲۰۰ نورون هست را به ما نشان می دهد. همان طور که میبینید شبکه به خوبی توانسته تابع  $\sin(2x+5)$  را تخمین بزند. همچنین داده بدون نویز این قسمت را در شکل TedadNoghat2.png در قسمت قبل بررسی کردیم. همانطور که مشاهده می کنید، مدل در این قسمت بسیار پیچیده تر است چرا که باید نویز را تشخیص دهد. هر چقدر نویز ایجاد شده ساده تر و نزدیک به صفر باشد، کار شبکه ساده تر است و هر چقدر نویز ایجاد شده پیچیده تر و بزرگ تر باشد، کار شبکه سخت تر است.

همچنین شکل Noise\_NoghatKam3.png مدل شبکه عصبی با MSE ۰.۳ را نشان می دهد که روی ۶۰۰ داده آموزشی آموزش داده شده و نویز با مقدار بزرگ در این نمونه وجود دارد. به طور کلی میبینیم که نویز کار شبکه را سخت تر کرده و مدل های ما نسبت به حالت قبل پیچیده تر شده اند.

#### 4. بخش چهارم:

در این قسمت می خواهیم تابع حاصل از خط خطی خودمان را با شبکه عصبی پیش بینی کنیم. شکل واقعی تابع در image5.png موجود است. دقت کنید که تعداد داده های تست و آموزش کم است. مدل شبکه عصبی با ۵ لایه همراه با ۱۰ نورون در هر لایه است. همچنین MSE این مدل برابر ۰.۰۶ است. شبکه نسبت به تعداد داده ها خوب عمل کرده. همچنین در Predict5.png مدل پیش بینی شده نمایش داده شده است. در بخش اول ما از پرسپترون چند لایه طبقه بندی شده استفاده کردیم، اما در این مسئله از پرسپترون چند لایه رگرسیون استفاده کردیم. در این مسئله می توان گفت خشم ما بیانگر پیچیدگی تابع خواهد بود و به همین دلیل هر چه خشم ما بیشتر باشد، پیچیدگی بیشتر بوده و اندازه شبکه نیز بیشتر خواهد بود. هر چقدر تعداد نقاط با پرش ناگهانی بیشتر بوده، دقت شبکه کمتر خواهد شد و پیش بینی خوبی نخواهد داشت و برای بهبودی آن، باید تعداد لایه ها و نورون ها را بیشتر کرد.

#### 5. بخش پنجم:

در این قسمت به کاربرد دسته بندی بیش از دو کلاس شبکه MLP پرداخته ایم.

دیتاست انتخاب شده در این مسئله، دیتاست USPS است که برای شناسایی ارقام دستنویس انگلیسی استفاده می شود. همانطور که مشخص است تعداد رقم ها از ۰ تا ۹ است، پس تعداد کلاس های ما ۱۰ خواهد بود. همچنین این مسئله را با کمک از کتابخانه ی `tensorflow` حل کرده. جالب است بدانید که کل فرایند آموزش و تست این مدل با استفاده از پرسپترون چند لایه ی طبقه بندی شده این کتابخانه به طور متوسط یک دقیقه طول میکشد. به ترتیب تعداد نوروں های قرار گرفته شده در همه لایه ها برابر است با ۱۰، ۱۲۸، ۱۲۸، ۲۵۶ که در ۳ لایه ابتدایی از تابع فعال سازی `Relu` استفاده شده و در لایه آخر از `softmax`.  
بهترین دقت به دست آمده در `accuracy5.png` نمایش داده شده که برابر ۹۴ درصد است.

## 6. بخش ششم:

در این قسمت به دیتاست موجود در بخش پنجم نويز اضافه کرده و آن را به عنوان داده آموزشی به شبکه داده و خواسته ایم که خروجی واقعی را برایمان پیش بینی کند برای حل این مسئله از پرسپترون چند لایه رگرسیون کتابخانه `sklearn` کمک گرفته همچنین دقت کنید که در این مدل از تمام داده های تست و آموزش استفاده کرده اما به دلیل طولانی شدن فرایند آموزش و تست شبکه، تعداد نوروں ها و لایه ها را کم و متعادل گرفته ایم. دقت کنید که با تعداد بالاتر هم تست شده، اما به دلیل خطا هنگام فرایند، سعی کردیم پیچیدگی را کم کنیم. در مشاهدات انجام شده شبکه تا حدود خوبی توانسته است که نويز را تشخیص دهد. همچنین دقت کنید که شبکه دارای ۴ لایه

پنهان با اندازه ۲۰ است. همچنین هر چه قدر نویز ایجاد شده بیشتر باشد، پیش بینی بدتر خواهد بود.