پارت اول:

دیتاست پروژه ی درخت تصمیم را با شبکه ی عصبی پیاده سازی کردم. Accuracy شبکه ی عصبی در این پروژه از درخت تصمیم کمتر است

دلایلی که ممکن است باعث شود دقت شبکه عصبی کمتر از دقت درخت تصمیم باشد، ممکن است به عوامل مختلفی برگردد. در زیر چند دلیل ممکن برای این موضوع آورده شده‌اند:

1.پیچیدگی مسئله:

- شبکه‌های عصبی معمولاً برای مسائل پیچیده‌تر و داده‌های بزرگ‌تر موثرتر هستند. اگر داده‌های شما نسبت به مسئله خیلی ساده باشند، شبکه عصبی ممکن است پیچیدگی اضافی داشته باشد که به دقت کلی کمک نکند.

2. مقدار داده:

- اگر مقدار داده شما محدود باشد، یعنی تعداد نمونه‌ها کم است، ممکن است شبکه عصبی دچار بیش‌برازش شود و بر روی داده‌های تست دقت کمتری داشته باشد.

3. پارامترهای مدل:

- تنظیم پارامترهای شبکه عصبی، مثل تعداد لایه‌ها، تعداد نورون‌ها و نرخ یادگیری، نقش مهمی در عملکرد مدل دارند. اگر این پارامترها به درستی تنظیم نشوند، مدل ممکن است به بهینه‌نشدن برسد.

4. اندازه داده:

- برخی از مدل‌ها، به ویژه شبکه‌های عصبی، برای دستیابی به عملکرد بهینه نیاز به داده‌های زیاد دارند. اگر داده‌ها کم باشند، شبکه عصبی به‌درستی آموزش نمی‌بیند.

5. انتخاب مدل مناسب:

- برای هر مسئله، مدل مناسبی وجود دارد. اگر درخت تصمیم برای مسئله شما بهترین عملکرد را داشته باشد، انتخاب شبکه عصبی ممکن است به جایزه نخورد.

6. پارامترهای آموزش:

- پارامترهای آموزش شبکه عصبی نیز نقش مهمی در دقت مدل دارند. مثلاً تعداد دوره‌ها (epoch) و اندازه دسته (batch size) می‌توانند بر عملکرد نهایی تأثیر بگذارند.

بررسی این عوامل و تنظیمات مختلف مدل می‌تواند به بهبود دقت شبکه عصبی کمک کند. همچنین، اگر دقت شبکه عصبی همچنان کمتر از درخت تصمیم باقی بماند، ممکن است مسئله شما برای یک مدل ساده‌تر قابل حل باشد و نیازی به پیچیدگی اضافی نداشته باشد.

پارت دوم:

1. تولید داده‌های آموزش:

- داده‌های آموزش با استفاده از تابع np.linspace برای مقادیر X تولید می‌شوند. توابع مختلفی برای تولید مقادیر متناظرمثل sin`، `cos`، `tan`، `log10 و ترکیبی از sin و log10 اعمال می‌شود. برای جلوگیری از مقادیر بی‌نهایت حاصل از تابع `log10`، یک پردازش ویژه برای جایگزینی مقدار مناسب (در این حالت 0) انجام می‌شود.

2. رگرسور MLP:

- یک رگرسور MLP با استفاده از MLPRegressor از scikit-learn ایجاد می‌شود. این مدل یک لایه مخفی با 20 نورون و سپس یک لایه مخفی دیگر با 10 نورون دارد. تابع فعال‌سازی به 'logistic' تنظیم شده است مدل با حداکثر 1000 تکرار آموزش می‌بیند.

3. تغییر اندازه داده:

- داده‌های آموزش برای آموزش بازشکل داده می‌شوند و یک دامنه وسیع‌تر از مقادیر ورودی X\_plot برای نمودار کشی ایجاد می‌شود.

4. نمودار کشی:

- زیرنمودارها برای هر تابع ایجاد می‌شوند. ستون چپ داده‌های آموزش و پیش‌بینی‌های MLP را نشان می‌دهد، در حالی که ستون راست داده‌های آزمون (خود تابع) و پیش‌بینی‌های MLP را نشان می‌دهد. میانگین مربعات خطا MSE برای هر تابع بر روی مجموعه آزمون محاسبه شده و در نمودارها نمایش داده می‌شود.

هرچه تعداد نورونها در دو لایه بیشتر باشد ، میزان MSE کمتر خواهد شد.

هر چه تعداد لایه ها کمتر باشد میزان MSE بیشتر است.

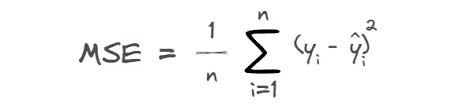
هر چه پیچیدگی تابع بیشتر باشد ، میزان MSE نیز بیشتر خواهد بود.

دلایل استفاده از : MSE

1. خطاهای بزرگ تر با وزن بیشتری در محاسبه معیار دخیل می‌شوند. خطاها را می‌سنجد به توان دو، که باعث می‌شود خطاهای بزرگتر تأثیر بیشتری داشته باشند.

2. مقیاس معیارها توسط میانگین مربعات توازن داده می‌شود. این معیار مقدار خطا را به صورت میانگین ارائه می‌دهد و از توازن برخی از اختلافات در مقیاس مقادیر واقعی و پیش‌بینی شده جلوگیری می‌کند.

هر چه مقدار MSE کمتر باشد، نشان‌دهنده عملکرد بهتر مدل است. در کد ارائه شده، مقدار MSE برای هر تابع بر روی مجموعه آزمون محاسبه شده و در نمودارها نمایش داده می‌شود.

فرمول: 

پارت سوم:

این بخش، همان بخش قبلی است با این تفاوت که با استفاده از حلقه ی for نقاطی را به صورت رندوم به نقاط اولیه تابع اضافه میکند، شبکه عصبی باز هم خروجی نزدیک به صحیح تولید میکند پس طبیعتا MSE تغییر چشمگیری نمیکند.