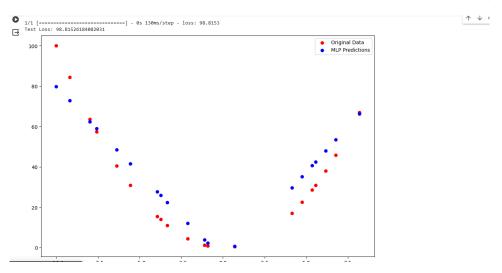
ابتدا چهار تابع از ساده تا پیچیده تعریف کردم و سپس تعدادی نقطه از روی این توابع تولید کردم و بخشی از این نقاط را به عنوان مجموعه آموزش دادم که بتواند تابع ورودی را تشخیص دهد. با استفاده از کتابخانه های موجود مانند matplotlib تابع خود را در کنار تابع تخمین زده شبکه تان رسم کردم و میزان خطای شبکه را نمایش دادم.

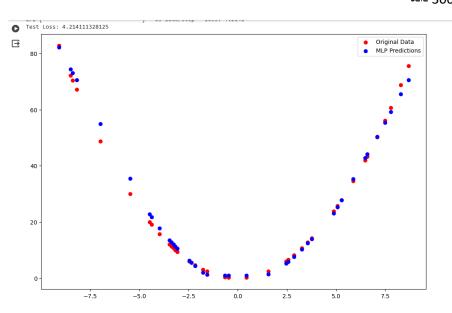
برای بهبود مدل پارامتر های مختلفی را برای هر تابع تغییر دادم.

برای مثال در تابع اول $f(x) = x^2$ تعداد داده های اولیه را 100 تا و برای دومین بار 300 داده در نظر گرفتیم که تفاوت بسیاری را نشان داد.

با 100 داده

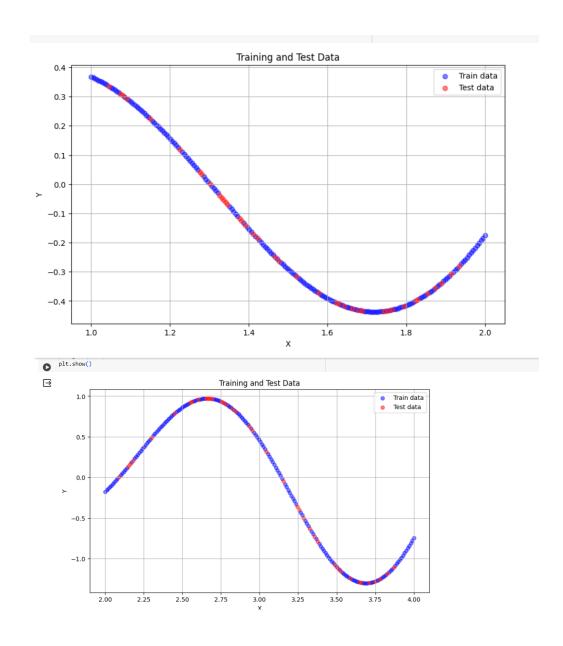


با 300 داده



برای دومین تابع به دلیل پیچیدگی زیادتر، تعداد لایه ها و مقدار نورون های هر لایه ها را زیادتر کردم و همچنین epoch را هم زیادتر در نظر گرفتم.

تابع سوم پیچیده تر بود به همین دلیل به بازه های کوچکتر تقسیمش کردم. برای مثال به جای اینکه تابع را در دامنه 1 تا 4 در نظر بگیرم. یکبار با دامنه 1 تا 2 و یکبار در دامنه 2 تا 4 در نظر گرفتم.



تابع چهارم هم تابع ساده ای بود که نیازی به پیچیدگی مدل نداشت. سپس به هر کدام از توابع مقداری نویز به صورت رندوم اضافه شده و در ایپاک بعدی مورد استفاده قرار می گیرد.

قسمت دوم MNIST:

تفاوت میان رفع نویز در داده های آموزشی و آز مایشی ممکن است به عوامل مختلفی بستگی داشته باشد. در ادامه به برخی از این عوامل اشاره میکنم:

میزان نویز در داده های آموزشی و آزمایشی :اگر میزان نویز در داده های آموزشی با داده های آزمایشی متفاوت باشد. متفاوت باشد.

معماری مدل :معماری شبکه عصبی و پارامترهای مدل میتواند تأثیر بسزایی در عملکرد رفع نویز داشته باشد. مدلی که برای آموزش استفاده میشود، ممکن است بر دقت رفع نویز در دادههای آموزشی و آزمایشی اثرگذار باشد.

نحوه اعمال نویز :اگر نویز به صورت مختلفی به دادههای آموزشی و آزمایشی اعمال شود، ممکن است نتایج متفاوتی حاصل شود. برای مثال، ممکن است نویزی که به دادههای آموزشی اعمال شده است با نوع نویز در دادههای آزمایشی متفاوت باشد.

اندازه دادهها :اندازه و تنوع دادهها نیز میتواند بر تفاوت در عملکرد رفع نویز در دادههای آموزشی و آزمایشی تأثیرگذار باشد. در صورتی که دادههای آزمایشی از نظر تنوع و توزیع با دادههای آموزشی متفاوت باشند، ممکن است نتایج متفاوتی حاصل شود.