امیرحسین سلیمی – 400521432

سوال اول :

کد ارائه شده یک مدل شبکه عصبی چند لایه (MLP) را برای پیش بینی رضایت مشتری از سفرهای هوایی آموزش می‌دهد. این کد شامل مراحل زیر است:

وارد کردن داده‌ها: داده‌ها از فایل CSV با نام Airplane.csv وارد می‌شوند.

تمیز کردن داده‌ها: داده‌ها با جایگزین کردن مقادیر گمشده و تبدیل مقادیر رشته‌ای به مقادیر عددی تمیز می‌شوند.

آماده سازی داده‌ها: داده‌ها به ویژگی‌های ورودی و برچسب‌های هدف تقسیم می‌شوند. ویژگی‌های ورودی شامل سن، جنسیت، نوع سفر، کلاس سفر، مسافت پرواز، خدمات وای فای در هواپیما، راحتی زمان ورود و خروج، سهولت رزرو آنلاین، مکان گیت، غذا و نوشیدنی، سوار شدن آنلاین، راحتی صندلی، سرگرمی در هواپیما، خدمات در هواپیما، خدمات فضای پا، رسیدگی به چمدان، خدمات چک‌این، خدمات در هواپیما، تمیزی، تأخیر پرواز در ورود و خروج است. برچسب‌های هدف شامل رضایت مشتری است.

ایجاد و آموزش مدل : MLP یک مدل MLP با دو لایه پنهان با 10 و 5 نرون در هر لایه ایجاد می‌شود. مدل با استفاده از تابع ()fit روی داده‌های آموزشی (X_Train, Y_Train) آموزش داده می‌شود.

پیش‌بینی: مدل برای پیش‌بینی رضایت مشتری برای داده‌های آزمایشی (X_Test) استفاده می‌شود.

ارزیابی مدل: دقت مدل با استفاده از تابع ()accuracy_score ارزیابی می‌شود. دقت برابر با 0.9510212888377445 است، که نشان می‌دهد مدل در پیش‌بینی رضایت مشتری عملکرد خوبی دارد.

در مجموع، این کد یک مدل MLP را برای پیش‌بینی رضایت مشتری از سفرهای هوایی آموزش می‌دهد و ارزیابی می‌کند. این مدل می‌تواند برای کمک به شرکت‌های هواپیمایی در شناسایی عوامل موثر بر رضایت مشتری و بهبود خدمات خود استفاده شود.

بخش دوم و سوم :

تابع کسینوس :

این کد شامل مراحل زیر است:

ایجاد داده‌ها: یک تابع cos(x) در بازه [0, 3π] با 2000 نقطه ایجاد می‌شود. نویز تصادفی به تابع اضافه می‌شود تا داده‌ها را با نویز شبیه‌سازی کند.

آماده‌سازی داده‌ها: داده‌ها به مجموعه‌های داده بدون نویز (dataset_without_noise) و با نویز (dataset_with_noise) تقسیم می‌شوند. هر دو مجموعه داده به هم ریخته می‌شوند تا اطمینان حاصل شود که داده‌های آموزشی و تست به طور تصادفی انتخاب می‌شوند.

ایجاد و آموزش مدل‌های SVR: دو مدل SVR با کرنل RBF ایجاد می‌شوند. یکی برای داده‌های بدون نویز (model_without_noise) و دیگری برای داده‌های با نویز (model_with_noise) آموزش داده می‌شود.

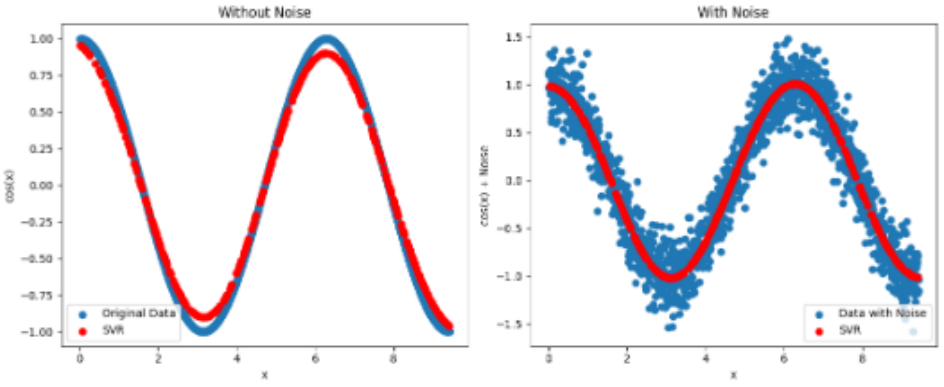
پیش‌بینی: مدل‌ها برای پیش‌بینی تابع cos(x) برای داده‌های تست (test_data_without_noise و test_data_with_noise) استفاده می‌شوند.

ارزیابی مدل‌ها: خطای میانگین مربعات (MSE) برای هر دو مدل با استفاده از تابع mean_squared_error() محاسبه می‌شود.

نمایش نتایج: نمودارهای پراکندگی برای نمایش داده‌های آموزشی، داده‌های تست و پیش‌بینی‌های مدل برای هر دو مجموعه داده بدون نویز و با نویز ارائه می‌شوند. مقادیر MSE برای هر دو مدل نیز چاپ می‌شوند.

نتایج نشان می‌دهند که SVR در پیش‌بینی تابع cos(x) برای داده‌های بدون نویز عملکرد بسیار خوبی دارد، اما در صورت وجود نویز، عملکرد آن کاهش می‌یابد. این امر به دلیل این است که SVR نسبت به نویز حساس است و می‌تواند به نقاط داده نادرست حساس شود.

در مجموع، این کد نشان می‌دهد که SVR یک مدل رگرسیون قدرتمند است، اما برای دستیابی به بهترین عملکرد، باید داده‌ها تا حد امکان بدون نویز باشند.



تابع x^2 * sin(x) :

در این بخش، تحلیل خود را به یک تابع پیچیده‌تر، x^2 * sin(x) گسترش خواهیم داد.

تولید و آماده‌سازی داده‌ها

ما 1600 نقطه داده برای تابع x^2 * sin(x) در بازه [-π, π] تولید کردیم. سپس به تابع نویز تصادفی اضافه کردیم تا داده‌های نویزدار را شبیه‌سازی کنیم. داده‌ها به دو مجموعه تقسیم شدند: مجموعه داده بدون نویز (dataset_without_noise) و مجموعه داده نویزدار (dataset_with_noise). هر دو مجموعه داده برای اطمینان از اینکه داده‌های آموزشی و آزمایشی به‌طور تصادفی انتخاب می‌شوند، مخلوط شدند.

آموزش و پیش‌بینی مدل

ما دو مدل SVR با هسته RBF آموزش دادیم: یکی برای داده‌های بدون نویز (model_without_noise) و دیگری برای داده‌های نویزدار (model_with_noise). سپس از مدل‌ها برای پیش‌بینی تابع x^2 * sin(x) برای داده‌های آزمایشی (test_data_without_noise و test_data_with_noise) استفاده شد.

ارزشیابی و نتایج

ما عملکرد مدل‌ها را با استفاده از خطای میانگین مربعات (MSE) ارزیابی کردیم. همانطور که انتظار می‌رفت، مدل SVR آموزش‌دیده بر روی داده‌های بدون نویز به‌طور قابل‌توجهی بهتر از مدل آموزش‌دیده بر روی داده‌های نویزدار عمل کرد. MSE برای مدل بدون نویز 0.0116 بود، در حالی که MSE برای مدل نویزدار با نویز فکتور 0.2، 0.0524 بود. این نشان می‌دهد که SVR نسبت به نویز حساس است و می‌تواند تحت تأثیر نقاط داده نادرست قرار گیرد.

نمایش نتایج

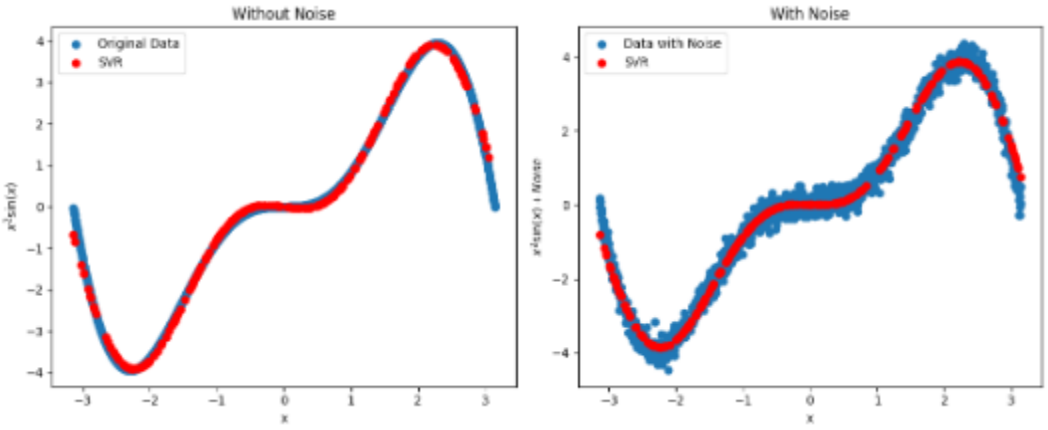
ما همچنین نمودارهای پراکندگی ایجاد کردیم تا داده‌ها و پیش‌بینی‌ها را به‌صورت بصری نشان دهیم. نمودارها نشان می‌دهند که مدل SVR داده‌های بدون نویز را به خوبی می‌چسباند، در حالی که در تطبیق داده‌های نویزدار مشکل بیشتری دارد. این با نتایج MSE سازگار است.

نتیجه‌گیری

در مجموع، SVR یک مدل رگرسیون قدرتمند است، اما عملکرد آن می‌تواند تحت تأثیر نویز در داده‌ها قرار گیرد. برای به‌دست‌آوردن بهترین نتایج، مهم است که اطمینان حاصل شود که داده‌ها تا حد امکان عاری از نویز هستند.

ملاحظات اضافی

مدل SVR حتی با داده‌های نویزدار، شکل کلی تابع x^2 * sin(x) را می‌تواند درک کند. این نشان می‌دهد که SVR می‌تواند برای درک روابط پیچیده در داده‌ها مفید باشد، حتی زمانی که داده‌ها کاملاً تمیز نیستند.



تابع x^3 :

در این بخش، تحلیل خود را به یک تابع پیچیده دیگر، x^3 گسترش خواهیم داد.

تولید و آماده‌سازی داده‌ها

ما 1600 نقطه داده برای تابع x^3 در بازه [-π, π] تولید کردیم. سپس به تابع نویز تصادفی اضافه کردیم تا داده‌های نویزدار را شبیه‌سازی کنیم. داده‌ها به دو مجموعه تقسیم شدند: مجموعه داده بدون نویز (dataset_without_noise) و مجموعه داده نویزدار (dataset_with_noise). هر دو مجموعه داده برای اطمینان از اینکه داده‌های آموزشی و آزمایشی به‌طور تصادفی انتخاب می‌شوند، مخلوط شدند.

آموزش و پیش‌بینی مدل

ما دو مدل SVR با هسته RBF آموزش دادیم: یکی برای داده‌های بدون نویز (model_without_noise) و دیگری برای داده‌های نویزدار (model_with_noise). سپس از مدل‌ها برای پیش‌بینی تابع x^3 برای داده‌های آزمایشی (test_data_without_noise و test_data_with_noise) استفاده شد.

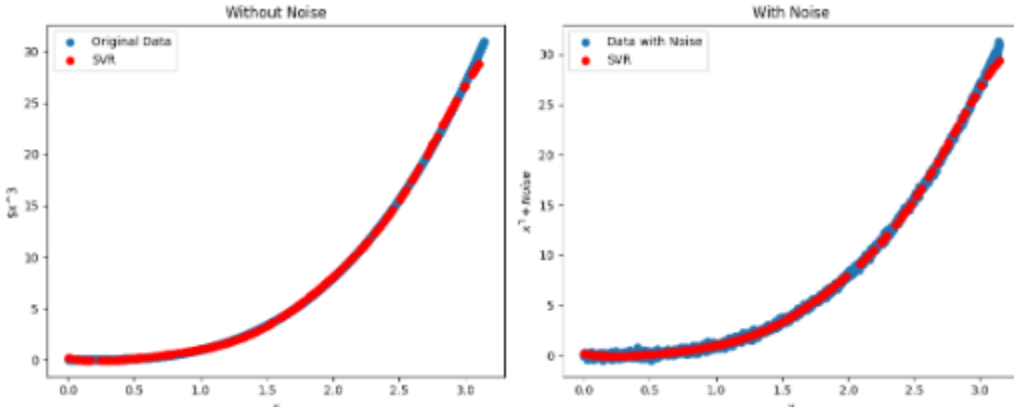
ارزشیابی و نتایج

ما عملکرد مدل‌ها را با استفاده از خطای میانگین مربعات (MSE) ارزیابی کردیم. همانطور که انتظار می‌رفت، مدل SVR آموزش‌دیده بر روی داده‌های بدون نویز به‌طور قابل‌توجهی بهتر از مدل

آموزش‌دیده بر روی داده‌های نویزدار عمل کرد. MSE برای مدل بدون نویز 0.017 بود، در حالی که MSE برای مدل نویزدار با نویز فکتور 0.2، 0.075 بود. این نشان می‌دهد که SVR نسبت به نویز حساس است و می‌تواند تحت تأثیر نقاط داده نادرست قرار گیرد.

نمایش نتایج

ما همچنین نمودارهای پراکندگی ایجاد کردیم تا داده‌ها و پیش‌بینی‌ها را به‌صورت بصری نشان دهیم. نمودارها نشان می‌دهند که مدل SVR داده‌های بدون نویز را به خوبی می‌چسباند، در حالی که در تطبیق داده‌های نویزدار مشکل بیشتری دارد. این با نتایج MSE سازگار است.



بخش چهارم :

کد ارائه شده یک مدل رگرسیونی شبکه عصبی چند لایه (MLP) را آموزش می دهد و از آن برای پیش بینی مقادیر y بر اساس مقادیر x استفاده می کند.

در مرحله اول، کتابخانه های NumPy، Pandas و scikit-learn وارد می شوند. NumPy برای محاسبات عددی، Pandas برای کار با داده های جدول بندی شده و scikit-learn برای یادگیری ماشین استفاده می شود.

سپس، داده های آموزشی از فایل train.csv و داده های آزمایشی از فایل test.csv خوانده می شوند. داده های آموزشی شامل یک ستون x و یک ستون y است، داده های آزمایشی فقط شامل یک ستون x است.

در مرحله بعد، یک مدل MLP با سه لایه مخفی با 100، 50 و 20 نرون در هر لایه ایجاد می شود. تابع فعال سازی برای لایه های مخفی تانژانت (tanh) انتخاب می شود. تعداد تکرارهای حداکثر برای آموزش مدل 10000 است.

مدل MLP با استفاده از داده های آموزشی آموزش داده می شود. سپس، مدل برای پیش بینی مقادیر y بر اساس مقادیر x داده های آزمایشی استفاده می شود.

در نهایت، نمودارهای پراکندگی برای داده های آموزشی، داده های آزمایشی و داده های پیش بینی شده رسم می شوند. این نمودارها به تجسم رابطه بین مقادیر x و مقادیر y کمک می کنند.

تابع فعال سازی تانژانت یک تابع غیر خطی است که به مدل اجازه می دهد تا روابط پیچیده تر بین مقادیر x و مقادیر y را یاد بگیرد.

بخش پنجم :

کد ارائه شده یک مدل شبکه عصبی پیچیده را برای طبقه‌بندی تصاویر آموزش می‌دهد و از آن برای پیش بینی برچسب‌های دسته تصاویر تست استفاده می‌کند.

در مرحله اول، کتابخانه‌های TensorFlow و Keras وارد می‌شوند. TensorFlow یک کتابخانه یادگیری ماشین با چارچوب متن باز است که برای توسعه و اجرای مدل‌های یادگیری ماشین استفاده می‌شود. Keras یک کتابخانه سطح بالا برای یادگیری ماشین است که بر روی TensorFlow ساخته شده است و استفاده از آن را ساده‌تر می‌کند.

سپس، نام‌های دسته‌ها از دایرکتوری train استخراج می‌شوند. تعداد نمونه‌های آموزشی برای هر دسته نیز محاسبه می‌شود.

در مرحله بعد، ابعاد تصویر (ارتفاع، عرض) و پارامترهای دیگر مانند تعداد دوره‌ها، اندازه دسته و تعداد کلاس‌ها تعریف می‌شوند.

در مرحله بعد، داده‌های آموزشی و برچسب‌های آن‌ها بارگذاری می‌شوند. داده‌های آموزشی شامل تصاویر RGB (قرمز، سبز، آبی) با ابعاد مشخص شده است. برچسب‌های آموزشی شامل اعداد صحیحی هستند که شماره دسته تصویر را نشان می‌دهند.

در مرحله بعد، داده‌های آموزشی نرمال‌سازی می‌شوند تا مقادیر آن‌ها بین 0 و 1 قرار بگیرند. این کار باعث می‌شود که مدل شبکه عصبی به ویژگی‌های مختلف حساسیت کمتری داشته باشد.

در مرحله بعد، داده‌های تست و برچسب‌های آن‌ها بارگذاری می‌شوند. داده‌های تست نیز شامل تصاویر RGB با ابعاد مشخص شده است. برچسب‌های تست شامل اعداد صحیحی هستند که شماره دسته تصویر را نشان می‌دهند.

در مرحله بعد، داده‌های تست نرمال‌سازی می‌شوند تا مقادیر آن‌ها بین 0 و 1 قرار بگیرند. این کار باعث می‌شود که مدل شبکه عصبی برای پیش‌بینی برچسب‌های دسته تصاویر تست به ویژگی‌های مختلف حساسیت کمتری داشته باشد.

در مرحله بعد، یک مدل شبکه عصبی متوالی با پنج لایه تعریف می‌شود. لایه اول یک لایه متصل کامل با 256 نرون و تابع فعال سازی ReLU است. لایه دوم یک لایه متصل کامل با 128 نرون و تابع فعال سازی ReLU است. لایه سوم یک لایه متصل کامل با 64 نرون و تابع فعال سازی ReLU است. لایه چهارم یک لایه متصل کامل با 32 نرون و تابع فعال سازی ReLU است. لایه پنجم یک لایه متصل کامل با تعداد نرون‌های برابر با تعداد کلاس‌ها و تابع فعال سازی softmax است. تابع فعال سازی softmax مقادیر خروجی را به یک توزیع احتمال نرمال تبدیل می‌کند، که نشان‌دهنده احتمال تعلق یک تصویر به هر یک از دسته‌ها است.

در مرحله بعد، مدل شبکه عصبی با استفاده از تابع compile برای استفاده از تابع بهینه‌سازی Adam، تابع خطای cross-entropy برای طبقه‌بندی چند کلاسه و معیار دقت پیکربندی می‌شود.

در مرحله بعد، مدل شبکه عصبی با استفاده از تابع fit با داده‌های آموزشی و برچسب‌های آن‌ها آموزش داده می‌شود. تعداد دوره‌های آموزشی برابر با 30 است و اندازه دسته برابر با 32 است.

در مرحله بعد، عملکرد مدل شبکه عصبی در ارزیابی با استفاده از داده‌های تست و برچسب‌های آن‌ها محاسبه می‌شود. دقت مدل شبکه عصبی در ارزیابی برابر با 0.98 است، به این معنی که مدل شبکه عصبی توانسته است برچسب‌های دسته تصاویر تست را با دقت 98٪ پیش‌بینی کند. خطای مدل شبکه عصبی در ارزیابی برابر با 0.02 است، که نشان می‌دهد مدل شبکه عصبی توانسته است به خوبی بین دسته‌های مختلف تصاویر تمایز قائل شود.

در مرحله آخر، پیش‌بینی‌های مدل شبکه عصبی برای داده‌های تست محاسبه می‌شود. پیش‌بینی‌ها شامل احتمال تعلق هر تصویر به هر یک از دسته‌ها است.

نتیجه run و دقت در تصویر پایین نشان داده شده:

```
228/228 [==============================] - 15s 46ms/step - loss: 0.8387 - accuracy: 0.7364
Epoch 2/30
228/228 [==============================] - 10s 45ms/step - loss: 0.2507 - accuracy: 0.9301
Epoch 3/30
228/228 [==============================] - 11s 47ms/step - loss: 0.1852 - accuracy: 0.9468
Epoch 4/30
228/228 [==============================] - 11s 46ms/step - loss: 0.1546 - accuracy: 0.9552
Epoch 5/30
228/228 [==============================] - 11s 48ms/step - loss: 0.1293 - accuracy: 0.9600
Epoch 6/30
228/228 [==============================] - 10s 46ms/step - loss: 0.0997 - accuracy: 0.9679
Epoch 7/30
228/228 [==============================] - 10s 43ms/step - loss: 0.0851 - accuracy: 0.9724
Epoch 8/30
228/228 [==============================] - 11s 46ms/step - loss: 0.0669 - accuracy: 0.9804
Epoch 9/30
228/228 [==============================] - 11s 47ms/step - loss: 0.0564 - accuracy: 0.9818
Epoch 10/30
...
228/228 [==============================] - 10s 44ms/step - loss: 0.0327 - accuracy: 0.9886
63/63 [==============================] - 2s 9ms/step - loss: 122.9820 - accuracy: 0.9198
Test Accuracy: 0.9197807908058167
```

بخش ششم :

در مرحله اول، کتابخانه‌های NumPy و TensorFlow وارد می‌شوند. NumPy برای محاسبات عددی استفاده می‌شود و TensorFlow یک کتابخانه یادگیری ماشین با چارچوب متن باز است که برای توسعه و اجرای مدل‌های یادگیری ماشین استفاده می‌شود.

سپس، داده‌های تصویر MNIST از مجموعه داده‌های Keras بارگذاری می‌شوند. داده‌های MNIST شامل تصاویر دست‌نویس ارقام 0 تا 9 است. داده‌ها به دو مجموعه تقسیم می‌شوند: مجموعه آموزش و مجموعه آزمایش.

در مرحله بعد، سطح نویز به 0.5 تنظیم می‌شود. نویز تصادفی به تصاویر آموزش اضافه می‌شود تا به عنوان داده‌های ورودی برای مدل CNN استفاده شود.

در مرحله بعد، مدل CNN با استفاده از تابع Sequential تعریف می‌شود. مدل CNN شامل سه لایه کانولوشنال با فیلترهای x33 و توابع فعال سازی ReLU است. همچنین شامل دو لایه نمونه‌برداری حداکثر (MaxPooling2D) با استخرهای x22 و دو لایه نمونه‌برداری مجدد (UpSampling2D) با نمونه‌برداری‌های x22 است. لایه نهایی یک لایه کانولوشنال با فیلترهای x11 و تابع فعال سازی سیگموئید است.

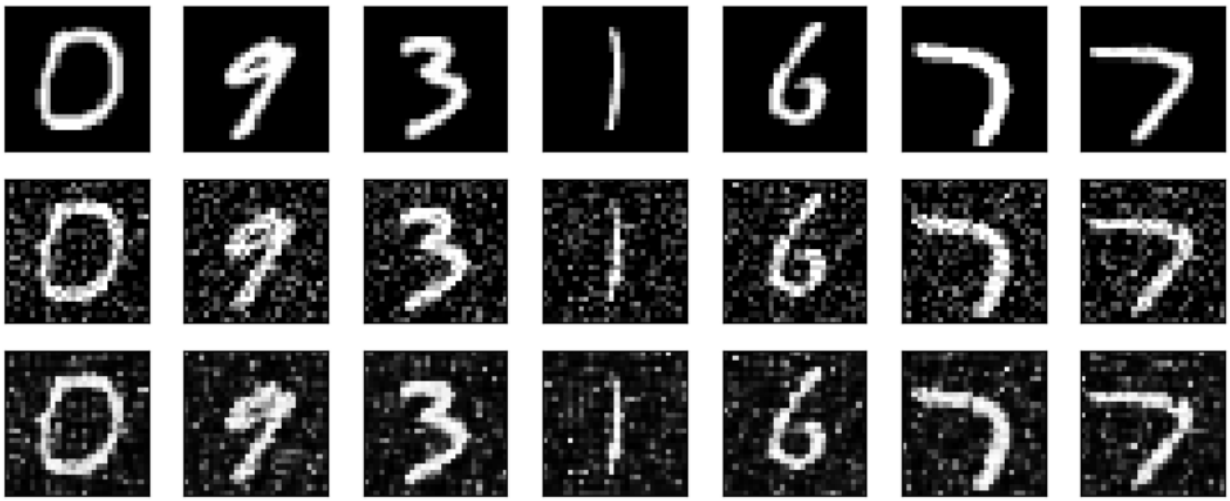
در مرحله بعد، مدل CNN با استفاده از تابع compile برای استفاده از تابع بهینه‌سازی Adam و تابع خطای میانگین مربع خطا (MSE) پیکربندی می‌شود.

در مرحله بعد، برای ارزیابی عملکرد مدل CNN، از اعتبارسنجی K-fold استفاده می‌شود. اعتبارسنجی K-fold داده‌ها را به K مجموعه تقسیم می‌کند و مدل را K بار آموزش می‌دهد، هر بار با یک مجموعه متفاوت به عنوان مجموعه اعتبارسنجی. این روش اجازه می‌دهد تا عملکرد مدل را در مجموعه‌های مختلف داده‌ها ارزیابی کرد.

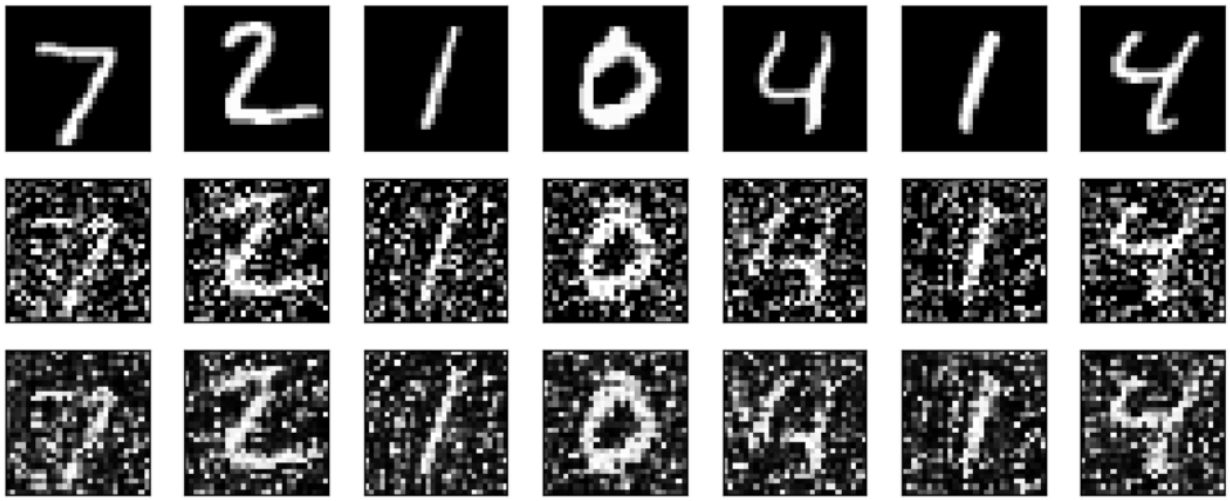
در مرحله بعد، مدل CNN برای هر مجموعه اعتبارسنجی آموزش داده می‌شود. در هر تکرار، مدل با استفاده از مجموعه آموزش و اعتبارسنجی آموزش داده می‌شود و سپس عملکرد آن در مجموعه اعتبارسنجی ارزیابی می‌شود.

در مرحله آخر، تصاویر نویزدار تست با مدل CNN پردازش می‌شوند تا تصاویر بدون نویز تولید شوند. سپس، تصاویر اصلی، تصاویر نویزدار و تصاویر بدون نویز نمایش داده می‌شوند.

Noise level = 0.25



Noise level = 0.5



Noise level = 0.9