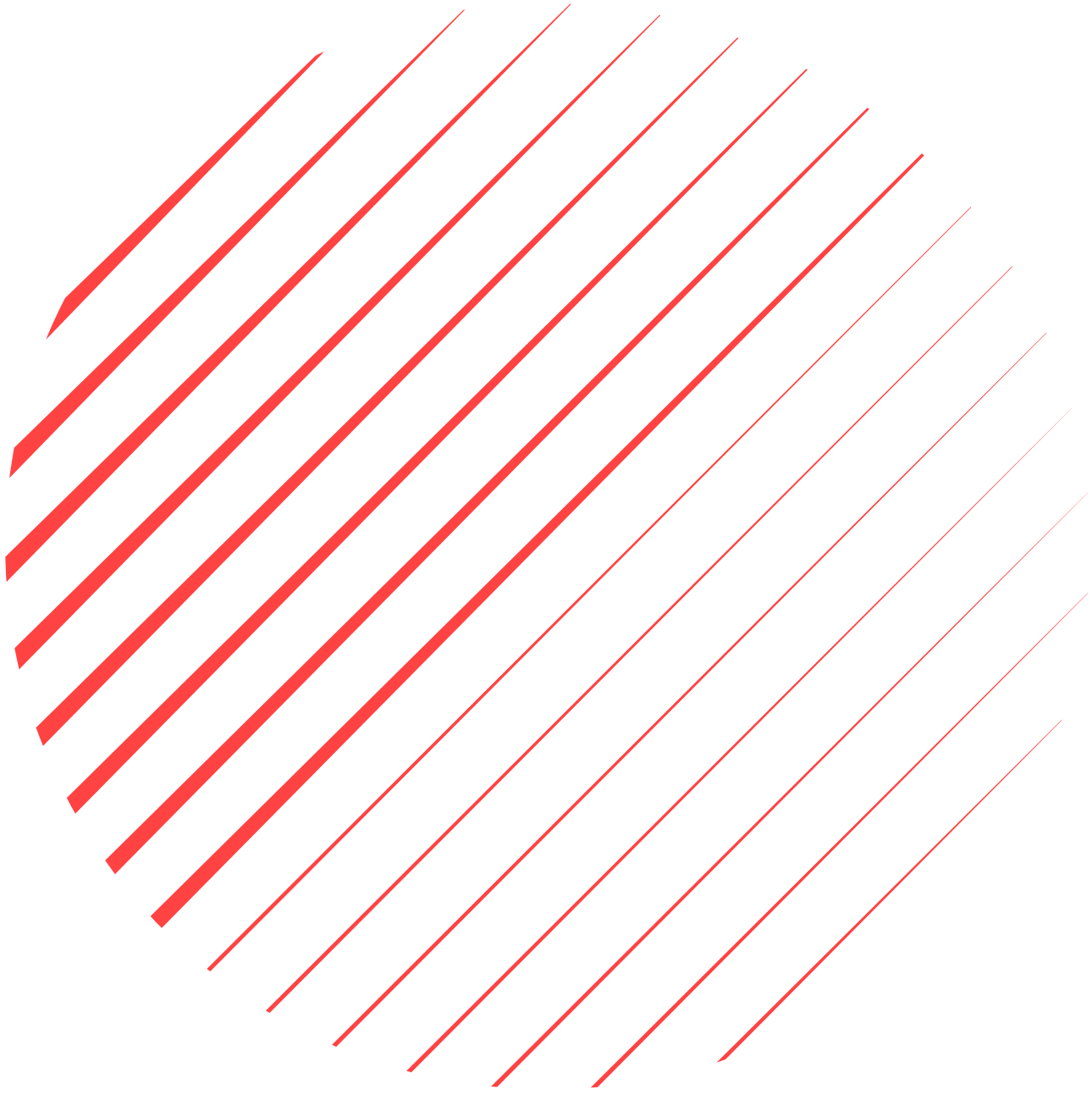
MINI PROJECT **#2**



FUNDAMENTAL OF INTELLIGENT SYSTEMS

Amir Mohammad Saffar
Dr. aliyarishorehdeli

Question 1

1.1

- اگر در مسئله طبقه‌بندی دوکلاسه، لایه‌های انتهایی شامل ReLUو سیگموید باشند، خروجی ReLU می‌تواند مقادیر بزرگ و نامحدود تولید کند که در سیگموید به 1 فشرده می‌شود، باعث اشباع سیگموید و از دست رفتن اطلاعات می‌شود. این ترکیب گرادیان‌ها را کاهش داده و یادگیری مدل را دشوار می‌کند.

راه‌حل :حذف ReLU از لایه‌های انتهایی و استفاده فقط از سیگموید در خروجی.

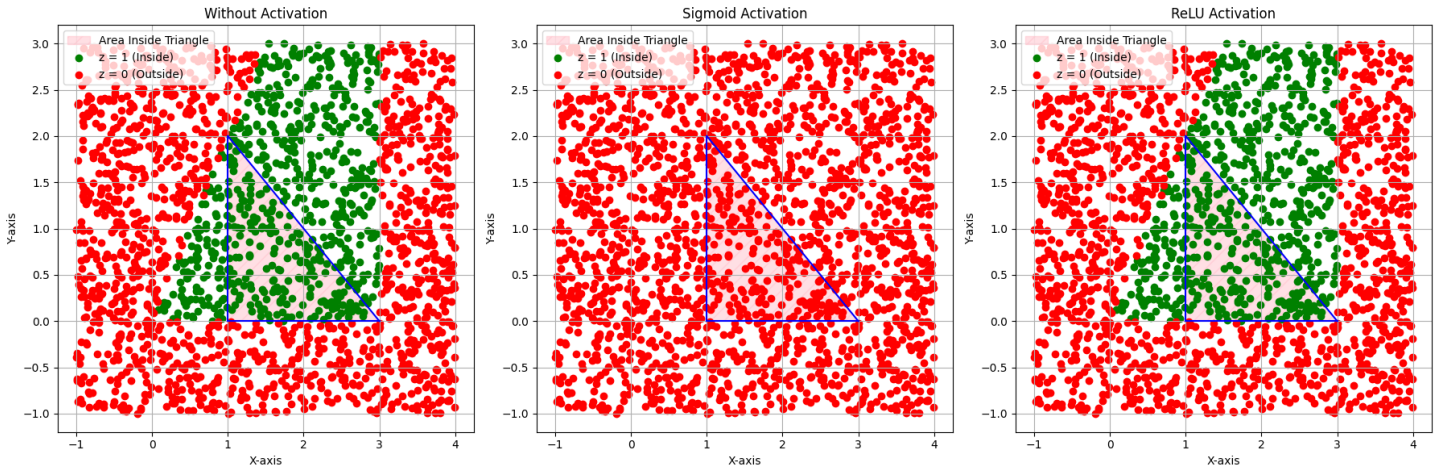
2.1

1. **مزیت اصلی ELU نسبت به ReLU:**
   - ELUدر مقادیر منفی، **خروجی‌های غیرصفر** تولید می‌کند، در حالی که ReLU تمام مقادیر منفی را به صفر نگه می‌دارد. این ویژگی از مشکل **مرگ نورون‌ها (Dead Neurons)** جلوگیری می‌کند.
2. **تغییرات تدریجی:**
   - درELU ، تغییرات خروجی برای مقادیر منفی به صورت نمایی و تدریجی انجام می‌شود، که باعث بهبود گرادیان‌ها در مقادیر کوچک می‌شود.
3. **مشکلDead ReLU:**
   - درReLU ، مقادیر منفی به صفر تبدیل می‌شوند، که می‌تواند باعث شود نورون‌ها دیگر یاد نگیرند (Dead Neurons).اما درELU ، به دلیل خروجی غیرصفر، این مشکل وجود ندارد.

برای ایکس های بزرگ تر از صفر مشتق بر با 1

برای ایکس های کوچکتر از صفر برابر $\alpha \times e^{x}$

3.1



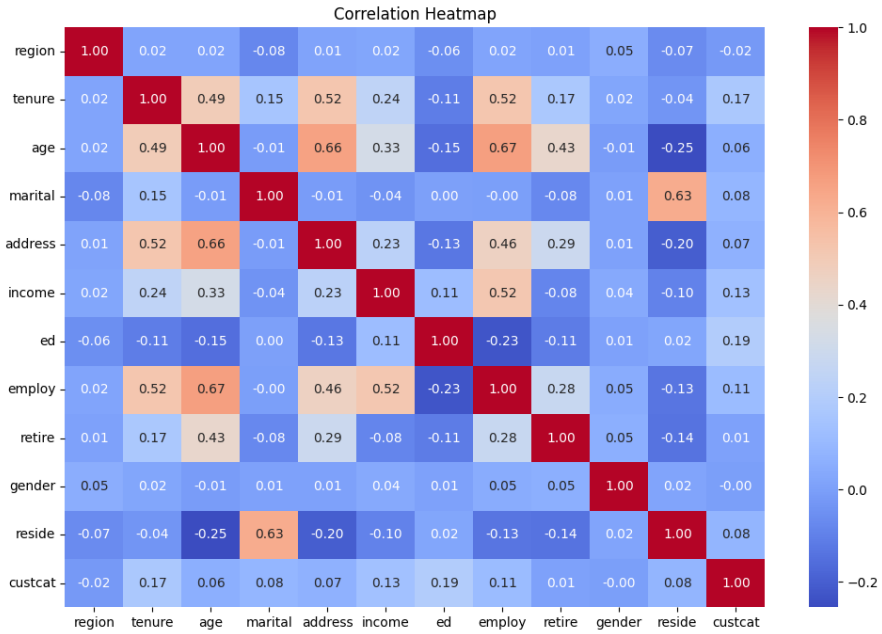
Question 2

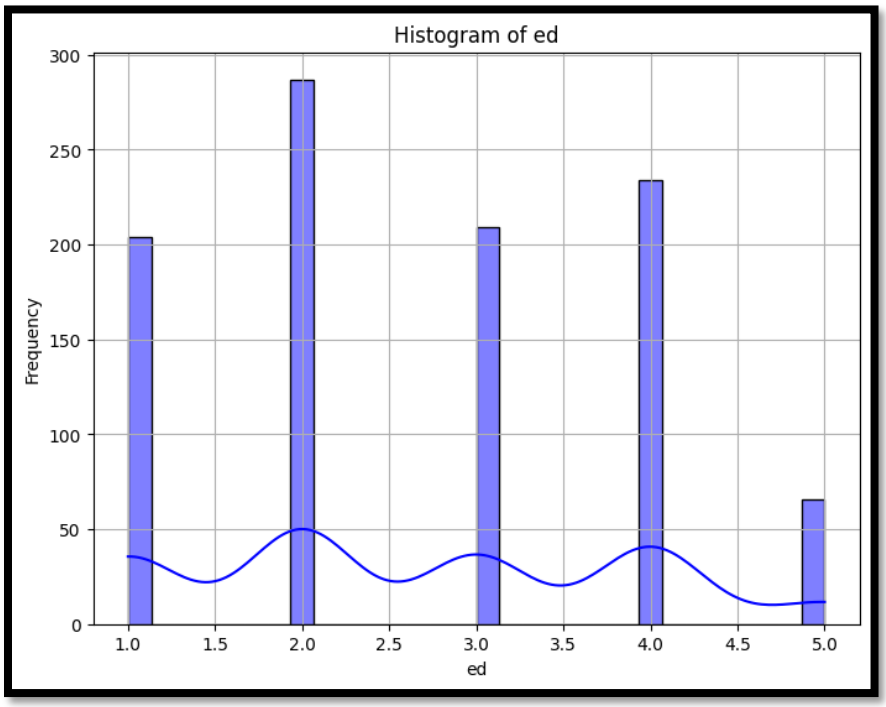
1.2

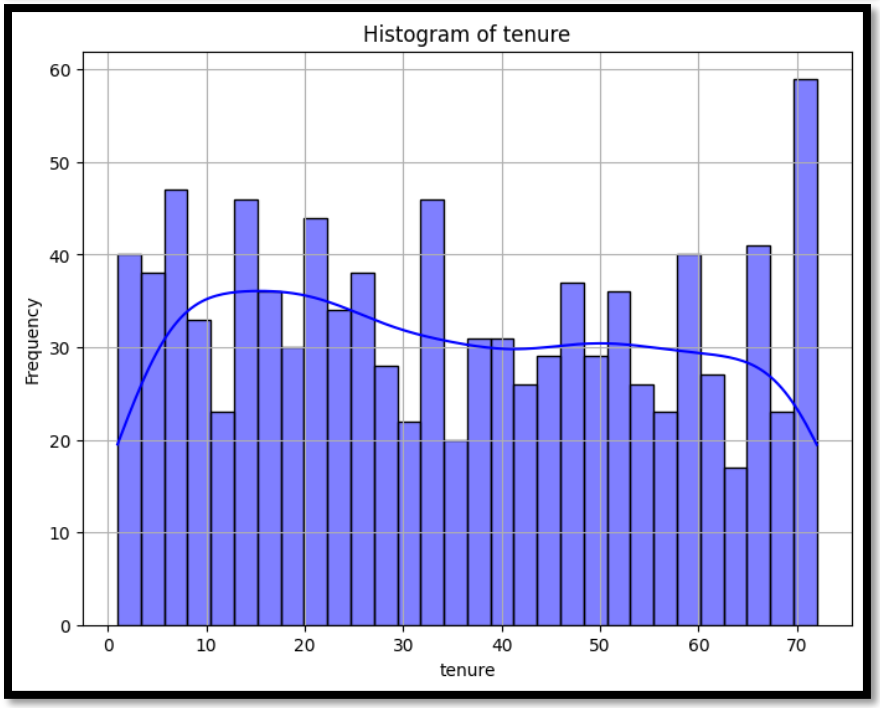
در داخل نوت بوک نوشته شده است

2.2

نمودار هیت مپ کرلیشن قابل مشاهده است و همچنین education و tenure دارای بیشترین کرلیشن با تارگت ما هستند که هیستوگرام آنها رسم شده است.




Histogram of ed
Frequency
ed
0
50
100
150
200
250
300
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


Histogram of tenure
Frequency
tenure
0
10
20
30
40
50
60
70

3.2

```
Training Set: 600 samples
Validation Set: 200 samples
Test Set: 200 samples
```

4.2

- تاثیر تعداد نورون‌ها :دو مدل با لایه پنهان (32 و 64 نورون) اجرا شد. عملکرد مدل با 32 نورون بهتر بود، ولی هر دو مدل دقت پایین‌تری داشتند، نشان‌دهنده پیچیدگی داده‌ها است.
- Batch Normalization : این لایه به بهبود دقت کمک کرد، زیرا سرعت همگرایی را افزایش داده و مدل را پایدارتر کرد. بهترین دقت در این حالت به حدود 37٪ رسید.
- Dropout :افزودن Dropout به مدل برای جلوگیری از بیش‌برازش انجام شد. مدل‌ها پایدارتر شدند ولی بهبود دقت محدود بود. بهترین دقت همچنان حدود 37.5٪ باقی ماند.
- L2-Regularization :اضافه کردن L2 Regularization باعث کاهش پیچیدگی مدل شد، اما دقت کلی تغییر قابل توجهی نداشت. دقت بهترین مدل حدود 39.5٪ ثبت شد.

: Adam

- عملکرد مدل با این بهینه‌ساز بهبود نسبی داشت و بهترین دقت اعتبارسنجی به 40.5%رسید.
- این بهینه‌ساز نوسانات کمتری در مقدار loss داشت و سریع‌تر به همگرایی رسید.

: RMSprop

- این بهینه‌ساز نیز عملکرد مشابهی با Adam داشت و بهترین دقت اعتبارسنجی آن حدود 41.9%بود.
- RMSpropبه دلیل ویژگی مقیاس‌بندی تطبیقی، در داده‌های پیچیده همگرا شد اما نتایج بسیار نزدیک به Adamبود.

5.2

```
Random Samples:
Index   True Label      RMSprop Prediction      Adam Prediction
61      3               3                       1
92      2               0                       0
58      1               0                       2
110     3               1                       1
6       1               3                       3
60      3               3                       1
147     3               3                       3
173     1               2                       2
13      2               2                       2
57      1               3                       3
```

6.2

دقت مدل‌ها:

- دقت مدل RMSprop: 38.5%
- دقت مدل Adam: 40.5%
- دقت مدل ترکیبی(Ensemble): 37.0%

نتیجه ترکیب:

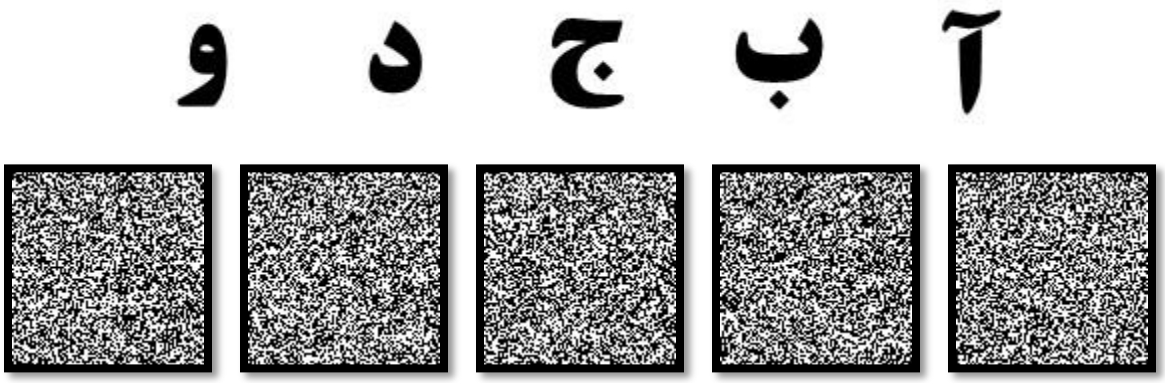
- مدل ترکیبی (Ensemble) نتوانست عملکرد بهتری نسبت به مدل‌های جداگانه داشته باشد.
- دقت آن از هر دو مدل RMSpropو Adamپایین‌تر بود.

دلایل احتمالی عدم بهبود:

- شباهت خطاها :اگر هر دو مدل خطاهای مشابهی داشته باشند، ترکیب مدل‌ها نمی‌تواند خطاها را جبران کند.
- کیفیت پیش‌بینی‌ها :اگر یکی از مدل‌ها بسیار ضعیف عمل کند، میانگین‌گیری احتمال‌ها یا ترکیب خروجی‌ها می‌تواند دقت کلی را کاهش دهد.
- اندازه داده‌ها :اگر داده‌های تست کوچک باشند، ترکیب مدل‌ها ممکن است به درستی نتایج را نشان ندهد.

Question 3

1.3



این تابع (convertImageToBinary) تصویر ورودی را به یک نمایش باینری تبدیل می‌کند. به صورت خلاصه عملکرد این تابع به شرح زیر است :

- *تصویر ورودی باز شده و ابعاد آن استخراج می‌شود .*
- *مقدار RGB هر پیکسل خوانده شده و مجموع شدت نور (intensity) آن محاسبه می‌شود .*
- *پیکسل‌هایی که شدت نور آن‌ها از یک آستانه (threshold) مشخص بیشتر باشند به رنگ سفید (با مقدار 1) و سایر پیکسل‌ها به رنگ سیاه (با مقدار 0) تبدیل می‌شوند .*
- *تصویر نهایی شامل رنگ‌های باینری سفید و سیاه بوده و لیستی از مقادیر باینری متناظر پیکسل‌ها بازگردانده می‌شود .*

نقاط ضعف و بهبود:

- *الگوریتم می‌تواند با استفاده از یک کتابخانه مانند NumPy به صورت برداری (vectorized) بازنویسی شود که عملکرد بهتری داشته باشد .*
- *به جای پردازش تک‌تک پیکسل‌ها، از عملیات ماتریسی برای پردازش تصویر استفاده شود .*
- *عملیات رسم تصویر (ImageDraw) اضافی به نظر می‌رسد و می‌توان آن را حذف کرد .*

2. این تابع (getNoisyBinaryImage) نویز تصادفی به یک تصویر اضافه کرده و آن را ذخیره می‌کند. عملکرد آن به صورت زیر است:

- تصویر ورودی باز شده و ابعاد آن استخراج می‌شود.
- یک مقدار نویز تصادفی برای هر پیکسل تولید شده و به مقادیر RGB پیکسل‌ها افزوده می‌شود.
- مقادیر RGB بعد از افزودن نویز محدود به بازه 0 تا 255 می‌شوند.
- تصویر نویزی ایجاد شده در فایل خروجی ذخیره می‌شود.

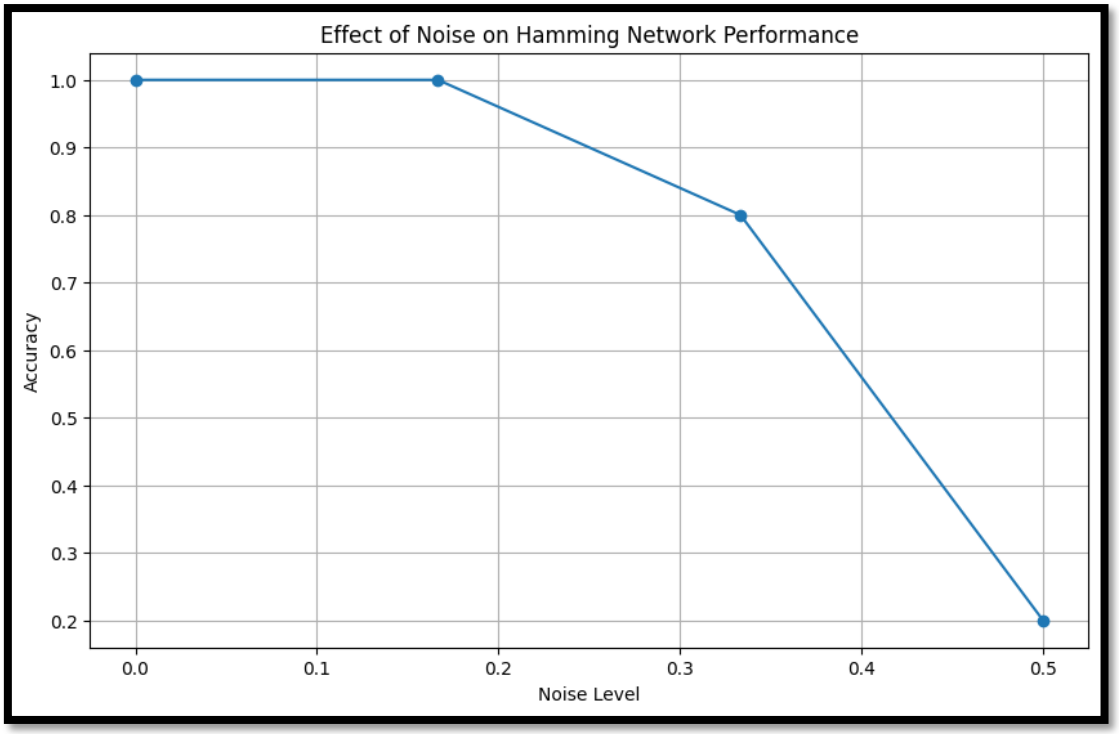
نقاط ضعف و بهبود:

- تعریف مقدار نویز (noise_factor) بسیار بزرگ است و ممکن است به اعوجاج شدید تصویر منجر شود.
- استفاده از کتابخانه NumPy برای اعمال نویز به صورت برداری می‌تواند سرعت و خوانایی کد را بهبود بخشد.
- افزودن نویز به تمام کانال‌های رنگی به صورت یکسان ممکن است به نویزی که رنگ‌های واقعی تصویر را بیشتر تخریب کند، منجر شود.

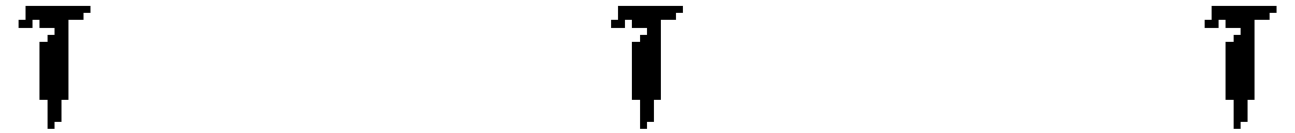
کد بهبود یافته در داخل دفترچه قرار داده شده است.

2.3

همانطور که مشاهده می کنید با افزایش مقدار نویز وارد شده به سیتم دقت در انتهای هر پارت بعد از مرتبه چهارم کاهش یافته است!


Effect of Noise on Hamming Network Performance
Accuracy
Noise Level
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.0
0.1
0.2
0.3
0.4
0.5

Original | Noisy (0.00) | Recreated

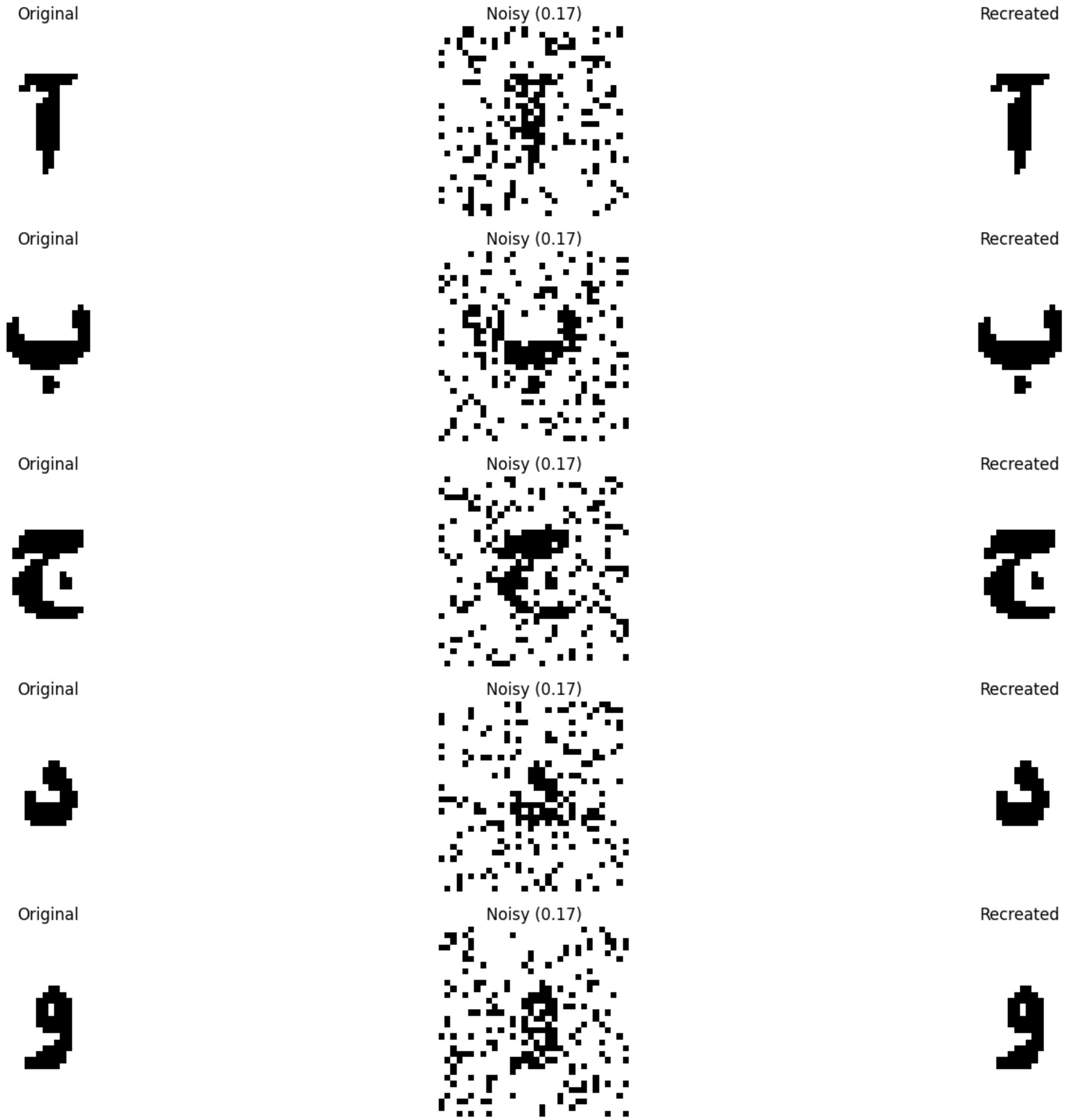


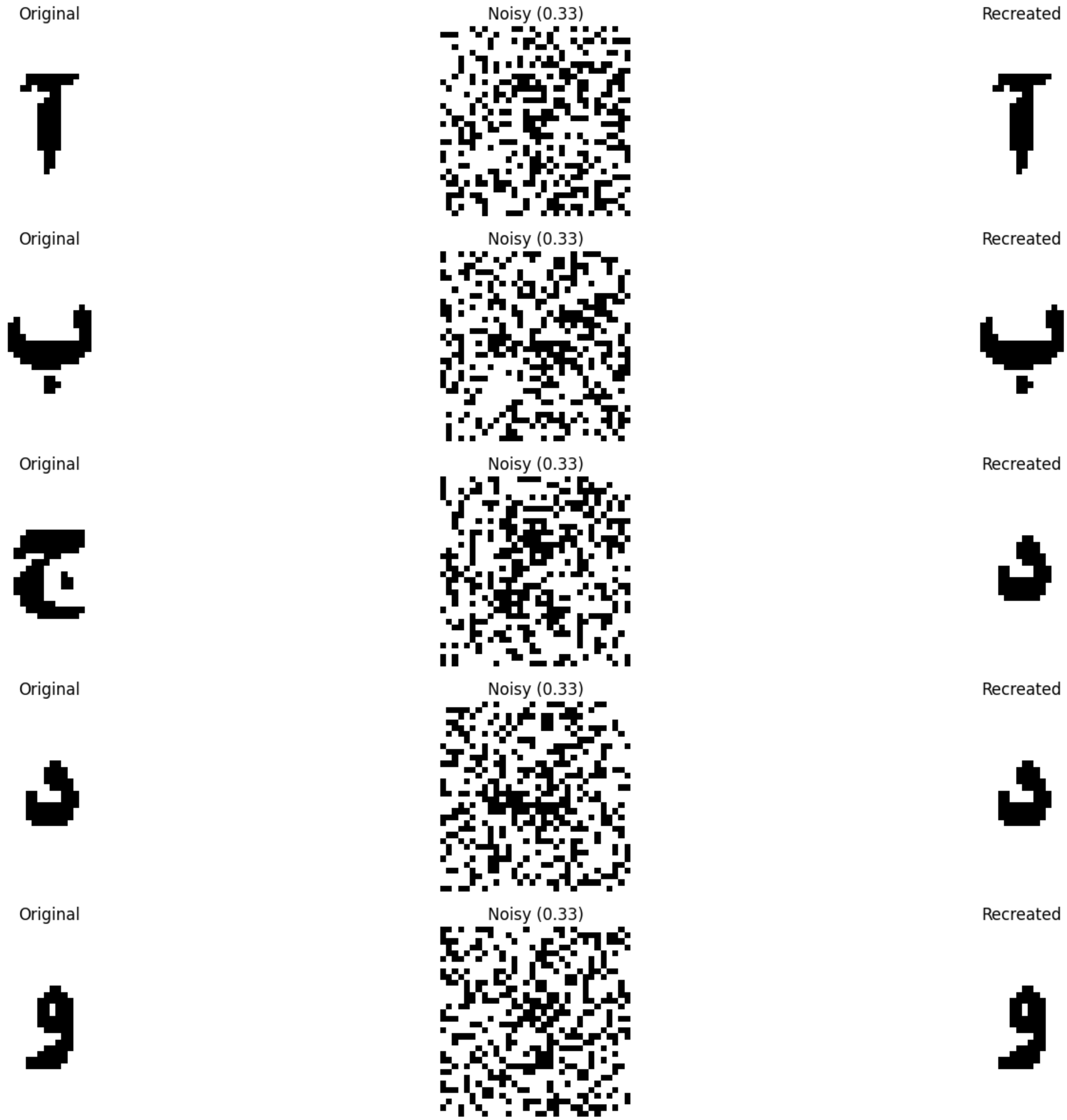
Original | Noisy (0.00) | Recreated

Original | Noisy (0.00) | Recreated

Original | Noisy (0.00) | Recreated

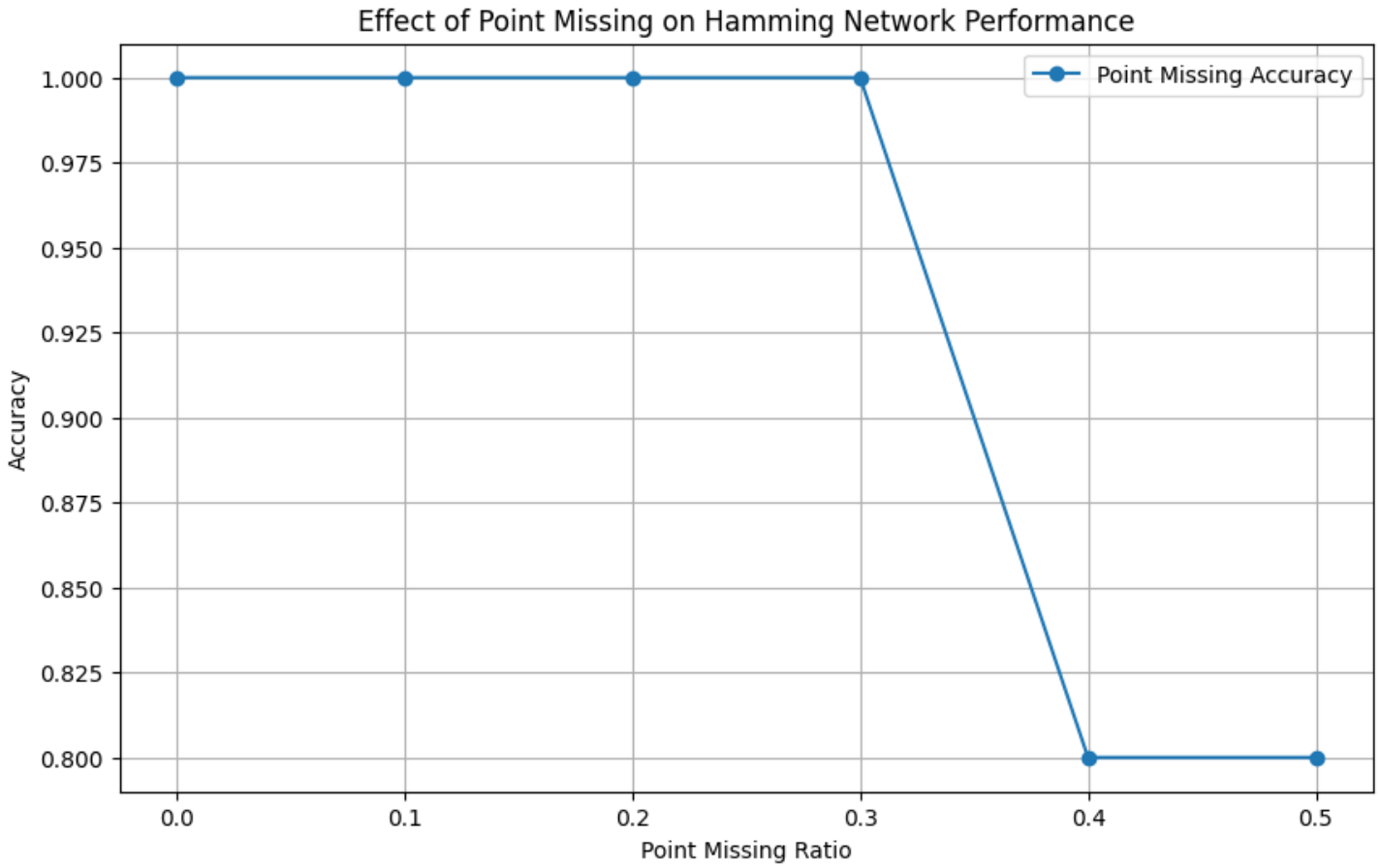
Original | Noisy (0.00) | Recreated


Original
Noisy (0.17)
Recreated
Original
Noisy (0.17)
Recreated
Original
Noisy (0.17)
Recreated
Original
Noisy (0.17)
Recreated
Original
Noisy (0.17)
Recreated



✓ ادامه خروجی ها در نوت بوک قرار داده شده است.

3.3


Effect of Point Missing on Hamming Network Performance
Point Missing Accuracy
Accuracy
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0.800
0.0
0.1
0.2
0.3
0.4
0.5
Point Missing Ratio

Original | Point Missing (0.50) | Recreated

Original | Point Missing (0.50) | Recreated

Original | Point Missing (0.50) | Recreated

Original | Point Missing (0.50) | Recreated

Original | Point Missing (0.50) | Recreated

- ✓ *زمانی که 30٪ تا 50٪ از پیکسل‌ها حذف شوند، عملکرد شبکه به شدت کاهش می‌یابد. برای مقادیر بالاتر از 50٪، احتمال بازسازی صحیح تصاویر بسیار پایین است .*
- ✓ *از دلایلی که میتوان به آنها اشاره کرد که سبب این امر می شوند می توان به موارد زیر اشاره کرد:*

از بین رفتن اطلاعات کلیدی:

- *زمانی که تعداد زیادی از پیکسل‌ها حذف می‌شوند، الگوی کلی تصویر به‌طور کامل از بین می‌رود و شبکه قادر به شناسایی آن نیست.*

وابستگی شبکه به الگوهای ذخیره‌شده:

- *شبکه همینگ برای بازیابی الگوهای نزدیک به داده‌های ذخیره‌شده طراحی شده است و اگر نویز یا حذف اطلاعات بسیار شدید باشد، این نزدیکی از بین می‌رود.*

برای بهبود عملکرد شبکه در مواجهه با نسبت‌های بالای Point Missing*، می‌توان از راهکارهای زیر استفاده کرد:*

استفاده از تکنیک‌های پیش‌پردازش

1. درون‌یابی داده‌های گم‌شده:(Interpolation)
   - *می‌توان از الگوریتم‌های درون‌یابی مانند میانگین مقادیر پیکسل‌های همسایه برای بازسازی نقاط گم‌شده قبل از ورود به شبکه استفاده کرد.*

2. استفاده از الگوریتم‌های یادگیری ماشین برای تخمین نقاط گم‌شده:
   - *می‌توان از مدل‌های پیچیده‌تر مانند* Autoencoder *استفاده کرد تا داده‌های گم‌شده را بازسازی کند.*

در کل:

عملکرد شبکه همینگ معمولاً تا حدود 30%حذف داده‌ها (Point Missing) پایدار باقی می‌ماند. برای حذف‌های بیش از این مقدار، دقت شبکه به شدت افت می‌کند.

Question 4

1.4

1. بارگذاری و آماده‌سازی داده‌ها
   - مجموعه داده‌های California Housingبارگذاری می‌شود که ویژگی‌های اقتصادی و جمعیتی نواحی مختلف کالیفرنیا را شامل می‌شود.
   - ویژگی‌ها (X) و هدف (y) از داده‌ها استخراج می‌شوند.
   - ویژگی‌ها با استفاده از "StandardScaler" استانداردسازی می‌شوند تا مقادیر آن‌ها مقیاس‌بندی شده و مدل بهینه‌تر آموزش ببیند.
   - داده‌ها به دو بخش آموزشی (Train) و تست (Test) تقسیم می‌شوند تا بتوان عملکرد مدل را روی داده‌های دیده‌نشده ارزیابی کرد.

2. تعریف لایهRBF
   - کلاس "RBFLayer" یک لایه سفارشی تعریف می‌کند که بر اساس تابع پایه گوسی Radial Basis Functionعمل می‌کند.
   - پارامتر unitsمشخص می‌کند چند مرکز RBF داریم.
   - پارامتر gammaتنظیم‌کننده حساسیت RBF به فاصله است.
   - در روش build، وزن‌های مربوط به مراکز (Centers) به عنوان پارامترهای قابل یادگیری تعریف می‌شوند.
   - در روش call، فاصله هر ورودی با مراکز محاسبه و خروجی گوسی به‌عنوان ویژگی‌های جدید تولید می‌شود.

3. ساخت مدل با لایهRBF
   - مدل با استفاده از Keras تعریف می‌شود:
   - لایه ورودی برای داده‌ها.
   - لایه RBF با 10 واحد برای مراکز RBF که ویژگی‌های ورودی را به خروجی گوسی تبدیل می‌کند.
   - یک لایه Dense خطی برای پیش‌بینی مقدار هدف (قیمت خانه).
   - مدل کامپایل شده و با تابع خطای MSE و متریک MAE آموزش می‌بیند.
   - مدل برای 50 اپوک با اندازه بچ 32 آموزش داده می‌شود.

4. ساخت مدل متراکم (Dense) برای مقایسه
   - یک مدل مقایسه‌ای شامل:
   - دو لایه Dense با 64 نورون و فعال‌سازیReLU .
   - یک لایه Dense خطی برای خروجی.
   - این مدل نیز با تنظیم مشابه MSE و MAE کامپایل و آموزش داده می‌شود.

5. ارزیابی و مقایسه مدل‌ها
   - هر دو مدل روی داده‌های تست ارزیابی می‌شوند و:
   - Loss (MSE): میانگین خطای مربعی.
   - MAE: میانگین خطای مطلق گزارش می‌شود.
   - هدف این است که عملکرد دو مدل در پیش‌بینی قیمت خانه مقایسه شود و تأثیر استفاده از لایه RBF تحلیل گردد.

کد در نوت بوک قرار داده شده است.

7-2.4

```
Dense Model - Test Loss: 0.2721835970878601, Test MAE: 0.3460240066051483

The Dense model performed better based on Test Loss.
```

مدلRBF : در مسائل پیچیده‌تر یا زمانی که داده‌ها روابط غیرخطی دارند، ممکن است عملکرد بهتری داشته باشد.
مدلDense : معمولاً در مسائل با داده‌های بزرگ و توزیع یکنواخت، به دلیل تعداد بیشتر پارامترها و انعطاف‌پذیری بالاتر، عملکرد بهتری دارد.

در این کد، دو مدل رگرسیون طراحی و آموزش داده شده‌اند: یکی با لایه RBF به عنوان لایه پنهان و دیگری با لایه‌های .Dense هر دو مدل از MSEبه عنوان تابع از دست دادن و Adamبهینه‌ساز استفاده می‌کنند. پس از آموزش، مدل‌ها روی داده‌های تست ارزیابی شده و عملکرد آن‌ها Loss و MAE مقایسه می‌شود. معمولاً مدل Denseبه دلیل انعطاف‌پذیری بیشتر و تعداد پارامترهای بالاتر، در داده‌های بزرگ‌تر عملکرد بهتری دارد، در حالی که مدل RBF برای روابط غیرخطی مناسب‌تر است.