دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

پروژه دوم هوش محاسباتی

شبکه‌های عصبی

اعضای گروه:

علی ابراهیمی

محمد اسحق

ترم دوم سال تحصیلی ۱۴۰۲-۱۴۰۱

۱- دسته بندی تصاویر با شبکه عصبی

در این بخش به توضیح یک الگوریتم شبکه عصبی برای کار با تصاویر پرداخته‌ایم. این شبکه عصبی برای کلاس‌بندی تصاویر طراحی شده و از چندین بخش تشکیل شده است. در ادامه به توضیح هر یک از این بخش‌ها می‌پردازیم.

الف) کلاس‌های لایه‌های شبکه عصبی

کلاس Dense برای ساخت لایه‌های تماماً متصل (Fully Connected) در شبکه عصبی استفاده می‌شود. در این کلاس، توابع forward و backward برای پیش‌برد و عقب‌گرد داده‌ها در شبکه عصبی پیاده‌سازی شده‌اند.

کلاس ReLU که مخفف Rectified Linear Unit است، یک تابع غیرخطی است که در شبکه‌های عصبی به عنوان تابع فعال‌ساز استفاده می‌شود. این کلاس نیز دارای توابع forward و backward است.

کلاس Sigmoid یک تابع فعال‌ساز دیگر است که در شبکه‌های عصبی استفاده می‌شود. این کلاس نیز توابع forward و backward را دارد.

کلاس Softmax تابعی است که خروجی شبکه عصبی را به احتمالاتی در بازه ۰ تا ۱ تبدیل می‌کند. این کلاس نیز دارای توابع forward و backward است.

کلاس اصلی مدل، کلاس CustomModel برای ساخت شبکه عصبی از این لایه‌ها استفاده می‌کند. این کلاس شامل توابع forward، backward و predict است. همچنین از کلاس CategoricalCrossEntropyLoss برای محاسبه خطا استفاده می‌کند.

در این الگوریتم شبکه عصبی، مراحل آموزش و ارزیابی به شرح زیر است:

ب) مرحله آموزش

در ابتدا، داده‌های آموزشی با استفاده از تابع load_data بارگذاری می‌شوند.
مدل شبکه عصبی با استفاده از کلاس‌هایDense ، ReLU و Softmax ساخته می‌شود.
به تعدادی دور (epoch) که در متغیر epochs تعیین شده، مراحل آموزش انجام می‌شود:
قسمت ویژگی‌ها (feature_extractor) از شبکه ResNet ۳۴ استفاده می‌شود.
برای هر بسته (batch) از داده‌های آموزش:
ویژگی‌های تصاویر استخراج می‌شود.
مقادیر ویژگی‌ها و برچسب‌های داده‌ها به فرمت آرایه‌ی numpy تبدیل می‌شوند.
مقادیر برچسب‌ها به فرمت one-hot encoding تبدیل می‌شوند.
مقادیر ویژگی‌ها به مدل شبکه عصبی داده می‌شود و خروجی مدل بدست می‌آید.
خطای مدل با استفاده از تابع خطا (CategoricalCrossEntropyLoss) محاسبه می‌شود.

گرادیان‌های مربوط به خطا نسبت به وزن‌ها و بایاس‌های شبکه محاسبه می‌شود.
وزن‌ها و بایاس‌های شبکه با استفاده از بهینه‌ساز (SGD) بروزرسانی می‌شوند.

ج) مرحله ارزیابی

پس از آموزش مدل، ارزیابی آن بر روی داده‌های آموزش و تست انجام می‌شود.
برای این کار، تابع evaluate_model فراخوانی می‌شود.
در این تابع، مراحل زیر انجام می‌شود:
ویژگی‌های تصاویر داده‌ها استخراج می‌شود.
مقادیر ویژگی‌ها به مدل شبکه عصبی داده می‌شود و خروجی مدل بدست می‌آید.
پیش‌بینی‌های مدل با استفاده از تابع predict محاسبه می‌شوند.
دقت مدل با مقایسه پیش‌بینی‌ها با برچسب‌های واقعی محاسبه می‌شود.
در نهایت، دقت مدل بر روی داده‌های آموزش و تست گزارش می‌شود.

```
model = CustomModel(layers=[
    Dense(n_features, 20),
    ReLU(),
    Dense(20, n_classes),
    Softmax()
])
optimizer = SGD(learning_rate=0.001)

epochs = 10
```

خروجی این بخش:

Epoch 0: 100%|[redacted]| 500/500 [00:55<00:00, 9.06batch/s]

Epoch:0

Loss: 1.1528416880992287

Accuracy: 0.65

Epoch 1: 100%|[redacted]| 500/500 [00:56<00:00, 8.83batch/s]

Epoch:1

Loss: 1.107264053450553

Accuracy: 0.69

Epoch 2: 100%|██████████| 500/500 [00:58<00:00, 8.58batch/s]

Epoch:2

Loss: 1.0168684402727357

Accuracy: 0.71

Epoch 3: 100%|██████████| 500/500 [00:57<00:00, 8.74batch/s]

Epoch:3

Loss: 1.1743683407662224

Accuracy: 0.66

Epoch 4: 100%|██████████| 500/500 [00:57<00:00, 8.77batch/s]

Epoch:4

Loss: 0.9633784783228769

Accuracy: 0.76

Epoch 5: 100%|██████████| 500/500 [00:56<00:00, 8.89batch/s]

Epoch:5

Loss: 0.9767196244438135

Accuracy: 0.76

Epoch 6: 100%|██████████| 500/500 [00:54<00:00, 9.12batch/s]

Epoch:6

Loss: 1.06305163672138

Accuracy: 0.71

Epoch 7: 100%|| 500/500 [00:57<00:00, 8.75batch/s]

Epoch:7

Loss: 1.0624666066902326

Accuracy: 0.71

Epoch 8: 100%|| 500/500 [00:55<00:00, 9.04batch/s]

Epoch:8

Loss: 0.9690672075994303

Accuracy: 0.74

Epoch 9: 100%|| 500/500 [00:55<00:00, 8.93batch/s]

Epoch:9

Loss: 1.0961506555537968

Accuracy: 0.7

Evaluating training set: 100%|| 500/500 [00:56<00:00, 8.79batch/s]

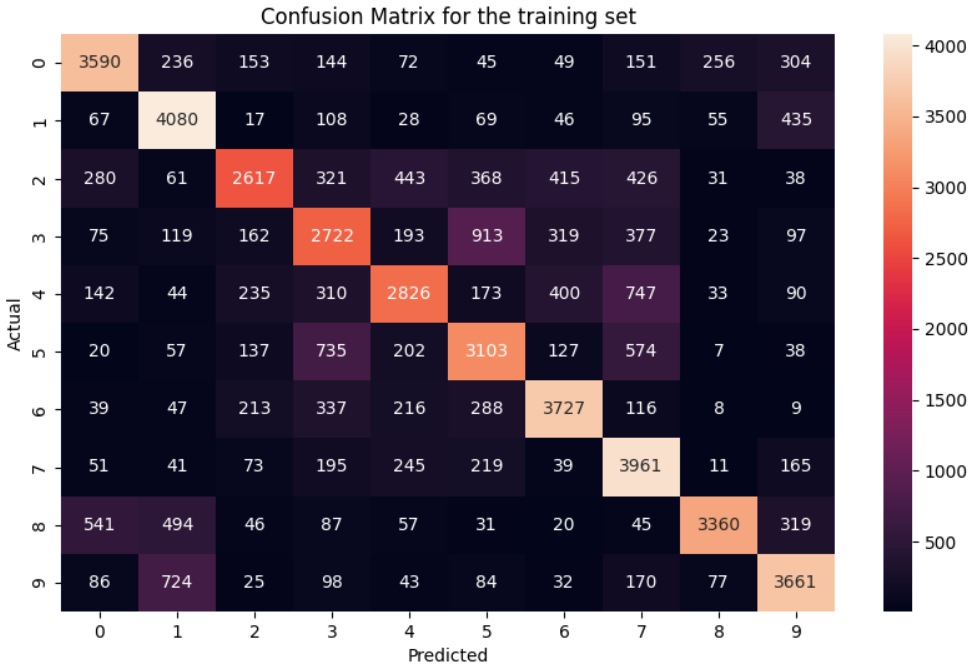
training Accuracy: 0.67294

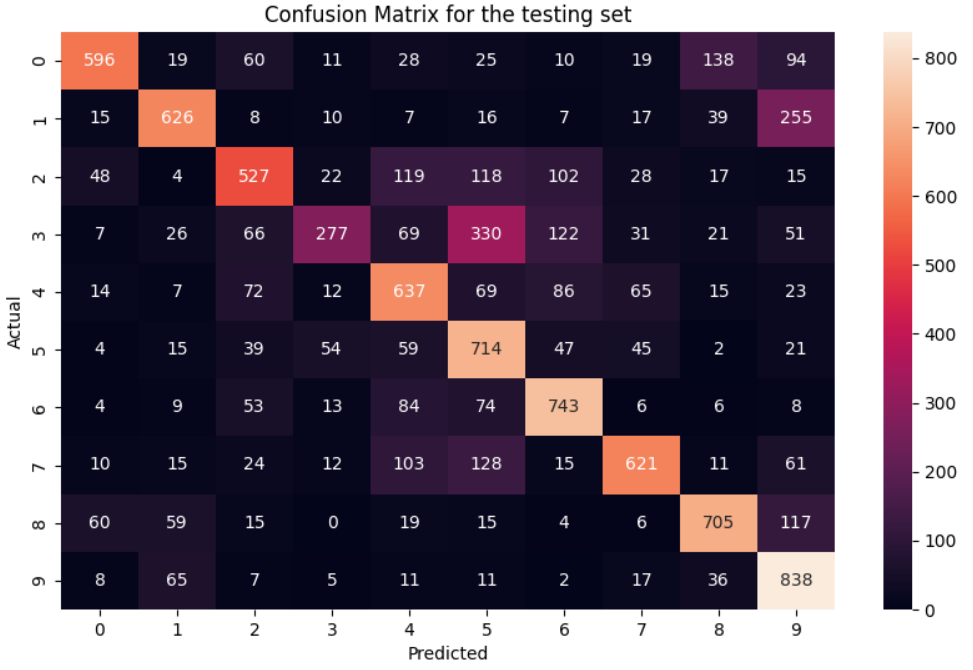
training F1 Score: 0.6719110511530144

Evaluating testing set: 100%|| 100/100 [00:11<00:00, 8.98batch/s]

testing Accuracy: 0.6479

testing F1 Score: 0.6461025536462485


Confusion Matrix for the training set
Actual
Predicted
0 1 2 3 4 5 6 7 8 9
3590 236 153 144 72 45 49 151 256 304
67 4080 17 108 28 69 46 95 55 435
280 61 2617 321 443 368 415 426 31 38
75 119 162 2722 193 913 319 377 23 97
142 44 235 310 2826 173 400 747 33 90
20 57 137 735 202 3103 127 574 7 38
39 47 213 337 216 288 3727 116 8 9
51 41 73 195 245 219 39 3961 11 165
541 494 46 87 57 31 20 45 3360 319
86 724 25 98 43 84 32 170 77 3661
4000 3500 3000 2500 2000 1500 1000 500


Confusion Matrix for the testing set
Actual
Predicted
0 1 2 3 4 5 6 7 8 9
596 19 60 11 28 25 10 19 138 94
15 626 8 10 7 16 7 17 39 255
48 4 527 22 119 118 102 28 17 15
7 26 66 277 69 330 122 31 21 51
14 7 72 12 637 69 86 65 15 23
4 15 39 54 59 714 47 45 2 21
4 9 53 13 84 74 743 6 6 8
10 15 24 12 103 128 15 621 11 61
60 59 15 0 19 15 4 6 705 117
8 65 7 5 11 11 2 17 36 838
800 700 600 500 400 300 200 100 0

۲- جستجوی معماری عصبی برای شناسایی الگو

هدف اصلی این الگوریتم تکاملی، بهینه‌سازی ساختار و پارامترهای شبکه عصبی برای حصول بهترین دقت در دسته‌بندی است. برای این منظور، از روش‌های انتخاب، بازترکیب (crossover) و جهش (mutation) استفاده می‌شود. در ادامه به توضیح روش‌های انتخاب شده برای هر یک از این مراحل پرداخته می‌شود:

انتخاب :(Selection) روش ترجیح‌پذیری ژنتیکی (Roulette Wheel Selection) یکی از روش‌های متداول برای انتخاب کروموزوم‌ها بر اساس نمره برازندگی است. در این روش، احتمال انتخاب هر کروموزوم متناسب با نمره برازندگی آن است. بنابراین، کروموزوم‌هایی با نمره برازندگی بالاتر احتمال بیشتری برای انتخاب به عنوان والدین دارند.

بازترکیب :(Crossover) روش یک نقطه‌ای (Single Point Crossover) می‌تواند برای بازترکیب دو کروموزوم استفاده شود. در این روش، یک نقطه تقاطع به صورت تصادفی انتخاب می‌شود و بخش‌های کروموزوم‌های والدین بعد از نقطه تقاطع با یکدیگر جابه‌جا می‌شوند تا دو فرزند جدید ایجاد کنند.

جهش :(Mutation) روش جهش تصادفی (Random Mutation) برای تغییر تک به تک ژن‌های کروموزوم‌های فرزند استفاده می‌شود. در این روش، با یک احتمال کم، هر یک از ژن‌های کروموزوم‌های فرزند به یک مقدار تصادفی جدید تغییر می‌کند.

با اجرای تعداد مشخصی نسل و انجام عملیات انتخاب، بازترکیب و جهش، کروموزوم با بالاترین نمره برازندگی به عنوان بهترین کروموزوم انتخاب می‌شود. سپس، مدل شبکه عصبی مرتبط با این کروموزوم ساخته و آموزش داده می‌شود. در نهایت، عملکرد مدل ساخته شده روی داده‌های آموزش و آزمون ارزیابی می‌شود.
این الگوریتم تکاملی برای تنظیم ابرپارامترهای شبکه عصبی به صورت خودکار انجام می‌شود و نیازی به تنظیم دستی ابرپارامترها نیست. این روش توانایی یافتن ترکیب‌های بهینه برای ساختار و پارامترهای شبکه عصبی را دارد و می‌تواند به افزایش دقت و کارایی مدل‌های یادگیری عمیق کمک کند.

```
population_size = 10
n_generations = 10
n_executions = 5
epochs = 5
```

بخشی از خروجی: (به علت طولانی بودن زمان اجرا از بقیه خروجی صرف نظر شد، اما پیش‌بینی آن نوشته شده است)

Epoch 0: 100%|[redacted]| 500/500 [00:37<00:00, 13.48batch/s]

Epoch:0

Loss: 1.2229674011961356

Accuracy: 0.62

Epoch 1: 100%|[redacted]| 500/500 [00:37<00:00, 13.24batch/s]

Epoch:1

Loss: 1.1026790604803565

Accuracy: 0.71

Epoch 2: 100%|[redacted]| 500/500 [00:36<00:00, 13.69batch/s]

Epoch:2

Loss: 0.9961234130109634

Accuracy: 0.71

Epoch 3: 100%|[redacted]| 500/500 [00:36<00:00, 13.73batch/s]

Epoch:3

Loss: 1.1470986988048018

Accuracy: 0.62

Epoch 4: 100%|[redacted]| 500/500 [00:36<00:00, 13.72batch/s]

Epoch:4

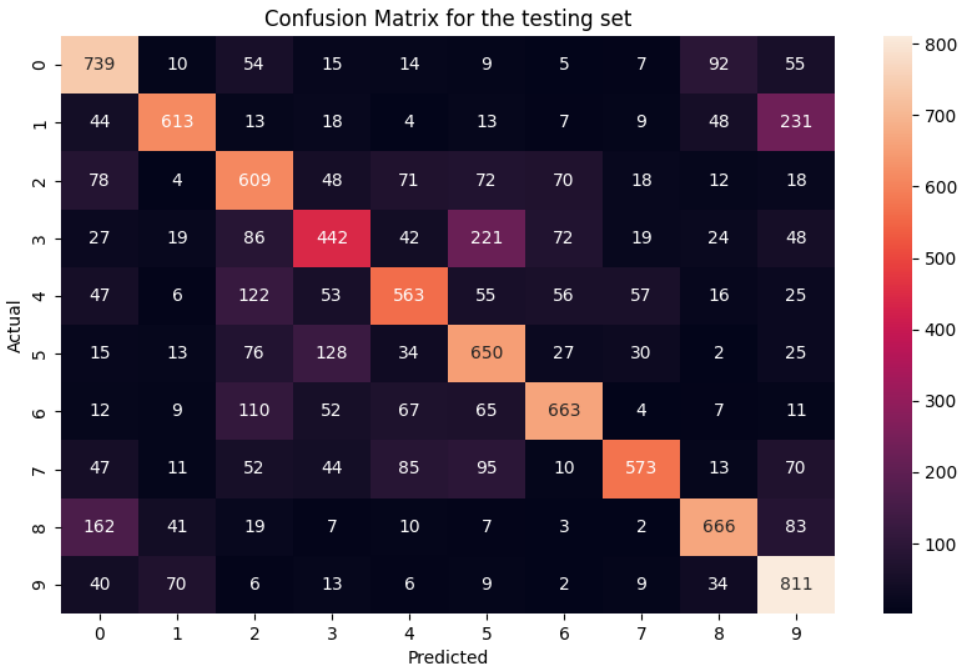
Loss: 0.9215087202789145

Accuracy: 0.78

Evaluating testing set: 100%|[redacted]| 100/100 [00:07<00:00, 12.77batch/s]

testing Accuracy: 0.6146

testing F1 Score: 0.6068783241650569



Epoch 0: 100%|[redacted]| 500/500 [00:36<00:00, 13.75batch/s]

Epoch:0

Loss: 1.2222072653138756

Accuracy: 0.64

Epoch 1: 100%|[redacted]| 500/500 [00:36<00:00, 13.75batch/s]

Epoch:1

Loss: 1.1824466944081626

Accuracy: 0.69

Epoch 2: 100%|[redacted]| 500/500 [00:36<00:00, 13.80batch/s]

Epoch:2

Loss: 1.0223847876656682

Accuracy: 0.68

Epoch 3: 100%|[redacted]| 500/500 [00:35<00:00, 14.07batch/s]

Epoch:3

Loss: 0.9162822174183812

Accuracy: 0.71

Epoch 4: 100%|[redacted]| 500/500 [00:35<00:00, 14.17batch/s]

Epoch:4

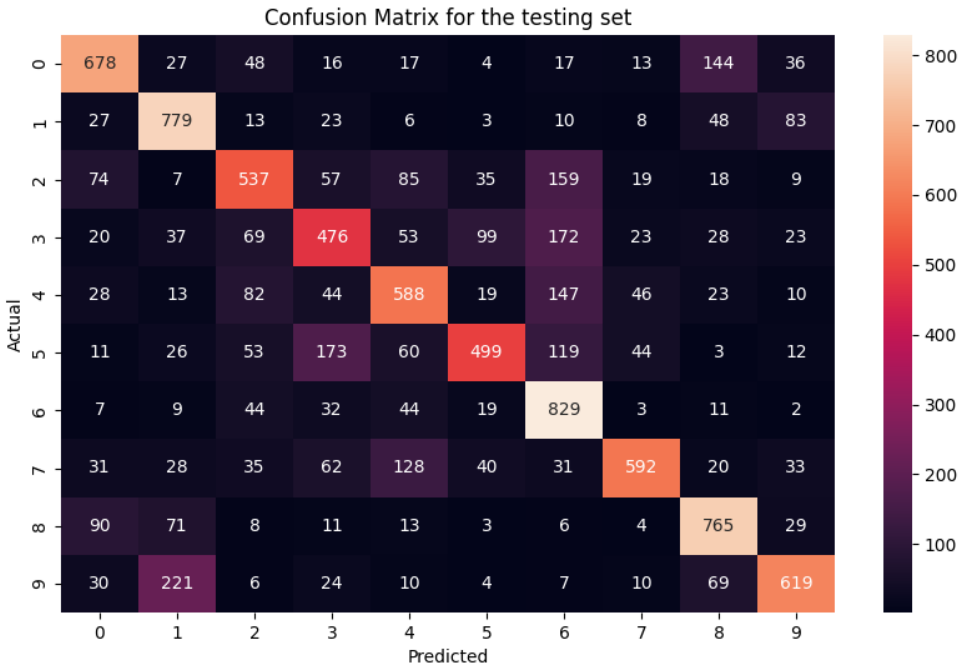
Loss: 1.1906661289624614

Accuracy: 0.63

Evaluating testing set: 100%|[redacted]| 100/100 [00:07<00:00, 13.29batch/s]

testing Accuracy: 0.6329

testing F1 Score: 0.632390879577148



Epoch 0: 100%|[redacted]| 500/500 [00:38<00:00, 13.10batch/s]

Epoch 1: 0%| | 0/500 [00:00<?, ?batch/s]

Epoch:0

Loss: 1.1778328849191526

Accuracy: 0.68

Epoch 1: 100%|[redacted]| 500/500 [00:38<00:00, 13.10batch/s]

Epoch:1

Loss: 1.1626777829259145

Accuracy: 0.59

Epoch 2: 100%|[redacted]| 500/500 [00:36<00:00, 13.70batch/s]

Epoch:2

Loss: 1.0099927561248911

Accuracy: 0.67

Epoch 3: 100%|██████████| 500/500 [00:35<00:00, 13.91batch/s]

Epoch:3

Loss: 0.8684333960664415

Accuracy: 0.71

Epoch 4: 100%|██████████| 500/500 [00:35<00:00, 13.93batch/s]

Epoch:4

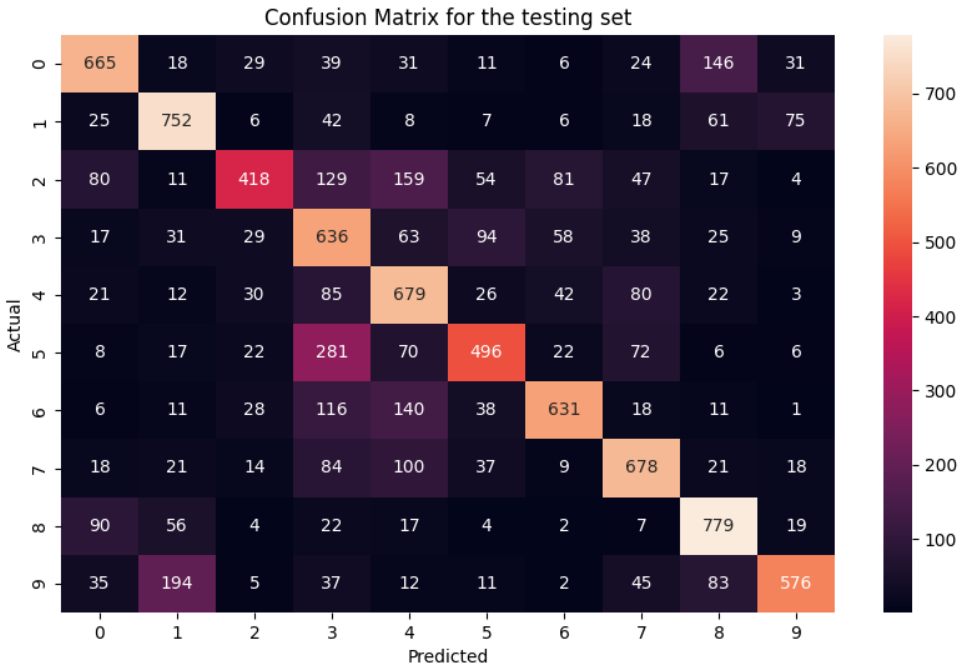
Loss: 1.0265763444844718

Accuracy: 0.61

Evaluating testing set: 100%|██████████| 100/100 [00:06<00:00, 14.48batch/s]

testing Accuracy: 0.6362

testing F1 Score: 0.6331908985639184



Epoch 0: 100%| | 500/500 [00:36<00:00, 13.56batch/s]

Epoch:0

Loss: 1.1292601932962127

Accuracy: 0.65

Epoch 1: 100%| | 500/500 [00:35<00:00, 14.07batch/s]

Epoch:1

Loss: 1.0707718145221776

Accuracy: 0.69

Epoch 2: 100%| | 500/500 [00:34<00:00, 14.48batch/s]

Epoch:2

Loss: 0.9687249074845758

Accuracy: 0.74

Epoch 3: 100%|██████████| 500/500 [00:36<00:00, 13.86batch/s]

Epoch:3

Loss: 0.8382048836820809

Accuracy: 0.78

Epoch 4: 100%|██████████| 500/500 [00:34<00:00, 14.44batch/s]

Epoch:4

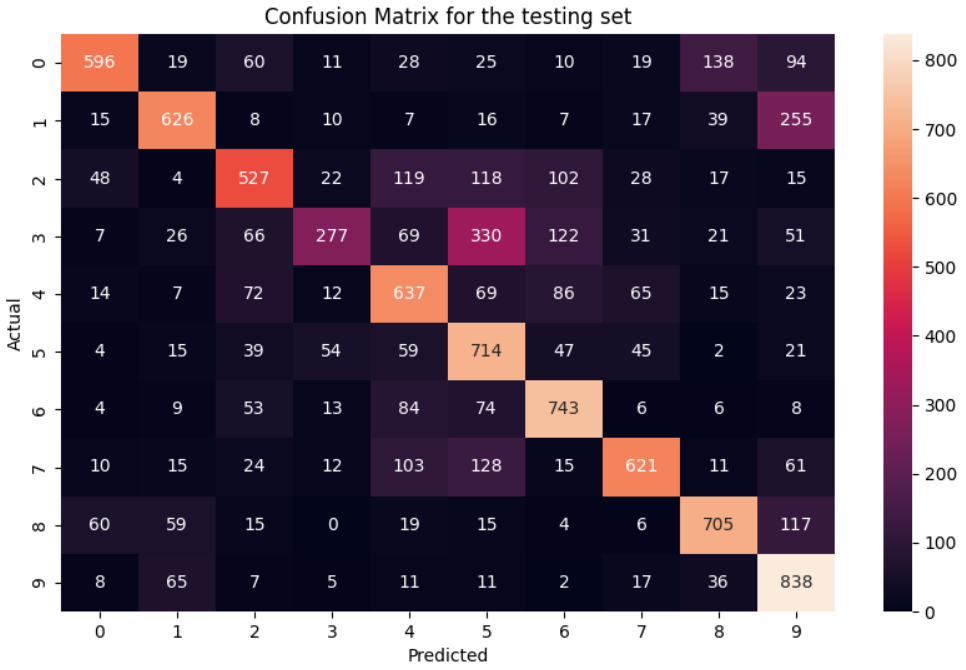
Loss: 0.9130915931969784

Accuracy: 0.72

Evaluating testing set: 100%|██████████| 100/100 [00:06<00:00, 14.58batch/s]

testing Accuracy: 0.631

testing F1 Score: 0.6308077035599131



با توجه به خروجی‌ها، می‌توان نکات زیر را بیان کرد:

ضرر (Loss) در هر دوره آموزش کاهش یافته است (به طور کلی). این به این معنی است که مدل در هر دوره بهینه‌تر شده و خطای پیش‌بینی کاهش یافته است.

دقت (Accuracy) در هر دوره آموزش افزایش یافته است (به طور کلی). این نشان می‌دهد که مدل در هر دوره بهتر عمل کرده و تعداد پیش‌بینی‌های درست افزایش یافته است.

دقت (Accuracy) در مجموعه آزمون بین ۰. ۶۱ تا ۰. ۶۴ تغییر کرده است. این نشان می‌دهد که مدل بر روی داده‌های جدید قادر به پیش‌بینی درست است.

امتیاز F۱ (F۱ Score) در مجموعه آزمون بین ۰. ۶۰ تا ۰. ۶۳ تغییر کرده است. این نشان‌دهنده توازن بین دقت و بازخورد در پیش‌بینی‌های مدل است.

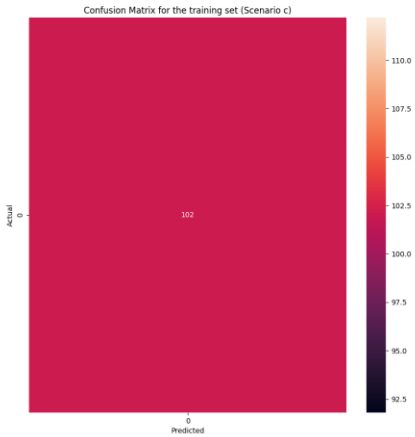
بر اساس این تحلیل، می‌توان پیش‌بینی کرد که اگر دوره‌های آموزش ادامه یابد، ممکن است ضرر کاهش یافته و دقت افزایش یابد. با این حال، باید مراقب Overfitting باشیم، که در این حالت مدل بیش‌ازحد به داده‌های آموزشی وابسته می‌شود و دیگر قادر به کلیت‌گیری بر روی داده‌های جدید نیست.

۳- نگاشت خود سازمانده

در این قسمت قصد داریم با استفاده از SOM یا همان شبکه کوهونن، داده‌ها را خوشه‌بندی کنیم. بدین منظور بعد از بارگزاری داده‌ها و استخراج ویژگی‌های آن‌ها به کمک SOM به هر یک از داده‌های موجود یک BMU (بهترین واحدی که این داده به آن تعلق داشته) را اختصاص داده و در نهایت خوشه‌بندی‌هایی این انجام داده را با برچسب‌های اصلی مقایسه کرده ایم.

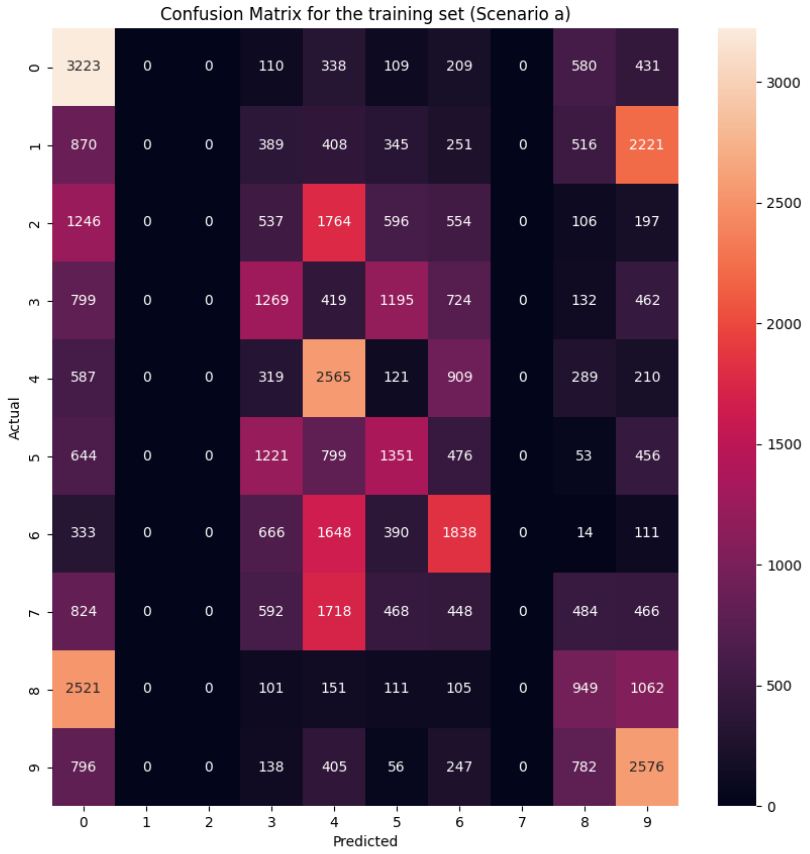
این الگوریتم به این صورت پیاده سازی شده است که ابتدا وزن‌هایی رندوم به هر یک از خوشه‌ها اختصاص میدهد. سپس در هنگام یادگیری، برای داده‌های ورودی بهترین واحد را بر حسب فاصله آن داده با هرکدام از واحدها انتخاب می‌کند و یک واحد به تعداد برچسب‌های موجود از این داده در واحد مورد نظر اضافه می‌کند. دقت شود که این کار صرفا برای پیشبینی‌های آینده مورد استفاده قرار می‌گیرد و تاثیری در یادگیری ندارد. در نهایت با توجه به داده‌ای که در واحد اضافه شده است، وزن‌های واحد مورد نظر اپدیت می‌شود تا تطابق بیشتری با داده وارد شده پیدا کند. همچنین ضریب یادگیری نیز در هر مرحله کاهش یافته تا رفته رفته میزان تغییرات الگوریتم کاهش بیابد.

برای بررسی الگوریتم استفاده شده به صورت ساده ابتدا داده‌هایی تنها با یک برچسب خاص را به ورودی می‌دهیم. اگر شبکه SOM نیز این داده‌ها را تنها در یک یا نهایتا دو خوشه قرار دهد و در ده خوشه موجود پخش نکند، به معنای این است که این شبکه کار خود را به درستی انجام می‌دهد.



*این بررسی در بین ۱۰۰۰ داده اولی که برچسب ۰ داشتند انجام شده است.

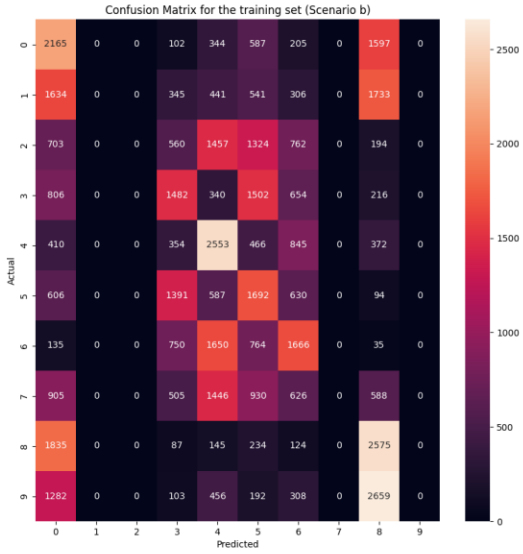
حال برای بررسی دقیق‌تر این مجموعه را در میان ۵۰۰۰۰ داده بررسی میکنیم (این الگوریتم تکمیل نشده و دارای دقت خوبی نیست)

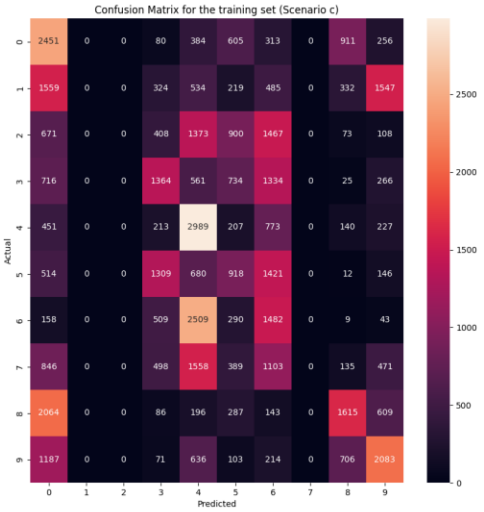


مشاهده می‌شود که داده‌ها تنها در ۷ خوشه دسته‌بندی شده اند. علت این امر این است که برخی دسته‌ها دارای اشتراک‌های زیادی بودند که باعث شده تمام آن‌ها صرفا در یک خوشه قرار بگیرند و بعضی از خوشه‌ها خالی بماند.

* البته دقت شود که ممکن است در این کلاس‌ها نیز داده‌هایی قرار گرفته باشند اما به علت اینکه الگوریتم پیش‌بینی برچسب‌ها به این صورت است که خروجی را خوشه‌ای در نظر میگیرد که بیشترین تعداد برچسب از نوع برچسب داده ورودی در آن قرار دارد.

در ادامه خروجی را برای حالت ب و پ مشاهده می‌کنید.





* شماره خوشه‌ها با برچسب های اصلی یکسان نیست. مورد قابل مشاهده این است که در هر ردیف، داده‌ها بیشتر به یک خوشه نسبت داده شده باشند که لزوما این شماره این خوشه با برچسب اصلی یکسان نخواهد بود. همانند ردیف ۶.

این مورد که داده‌ها در ردیف‌های ۵ و ۳ در خوشه های مختلفی پخش هستند نشان‌دهنده تشابه این داده‌ها با داده‌های خوشه‌های مختلف است (و نه یک خوشه خاص) که از آنجایی که مربوط به برچسب‌های سگ و گربه اند، این امر طبیعی است.