משימה 3:

1. הגרפים שמייצגים את השגיאה הממוצעת לאורך הepoch-ים :

תמונה שמכילה טבלה

התיאור נוצר באופן אוטומטי

ניתן לראות שהשגיאה של האימון הולכת וקטנה מהר יותר מהשגיאה של הtest מכיוון שיש overfitting לאחר 5 epoch-ים .

התמונות שהמודל סיווג לא כמו שצריך :

תמונה שמכילה טבלה

התיאור נוצר באופן אוטומטיהמספר ליד האות p מייצג את התחזית והמספר שליד האות a מייצג את תוצאה האמיתית לפי הlabels .

ניתן לראות שבחלק מהמקרים הדגימות שבהן המודל טעה באמת דומות ויזואלית למספר שהוא חזה שהן מייצגות עד כדי הבדלים של קווים קטנים.ניתן להסיק מכך שהמודל מסוגל לזהות מאפיינים כללים של הספרה ויודע להכריע על פיהם.

השגיאה הסופית ב *יוצאת :*

2)

תמונה שמכילה טבלה

התיאור נוצר באופן אוטומטי

ניתן לראות ששימוש בseeds שונים לא משנה דרמטית את התוצאות הסופיות.מכיוון שהשונות בין היא בערך (מספר מדויק ניתן לראות בכותרת ) בנוסף ממוצע השגיאה הסופית בין הseeds השונים הוא גם מאוד נמוך בערך 0.0007 . מכך אנחנו מסיקים שהמודל שלנו הוא יציב כלומר "חסין" מפני שינוי seed .

3)

|  |  |
| --- | --- |
|  |  |
|  |  |
| 0.0008608867950737477 | 0.0008117270994538558 |
| 0.0009031318088062108 | 0.0008439078916111612 |
| 0.0008872561268508434 | 0.0008455797003407496 |
| 0.0009235832986421883 | 0.0008470360174949747 |

בנוסף , חישבנו את הממוצע וסטיית התקן עבור ערכי  *שבטבלה והם :*

*std=3.766066947719082e-05, mean=0.0008662097970955074*

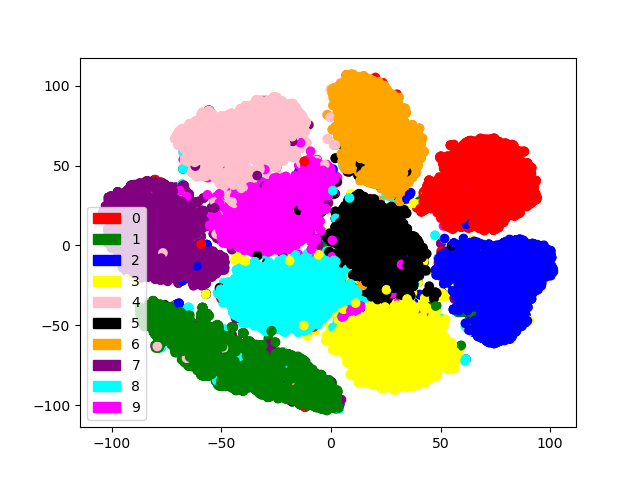
*ע"י מציאת הepoch עם שגיאת הולידציה הקטנה ביותר אנחנו מצליחים לתפוס את הרגע שבו המודל לא מאומן מידי (כלומר לא קורה overfitting) כך שעברנו כמות "אופטימלית" של epoch-ים .*

4)

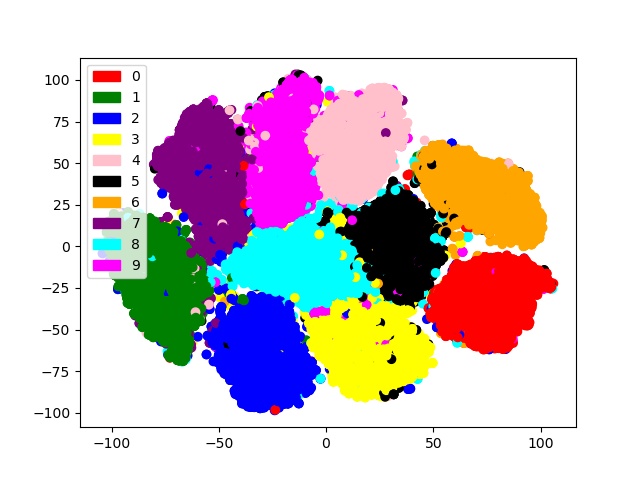
|  |  |  |  |
| --- | --- | --- | --- |
| Hidden\_size | Batch\_size | Learning\_rate | Test error |
| 200 | 50 | 0.001 | 0.006323 |
| 500 | 50 | 0.001 | 0.005258 |
| 600 | 50 | 0.001 | 0.00524 |
| 200 | 100 | 0.001 | 0.00347 |
| 500 | 100 | 0.001 | 0.0032 |
| 600 | 100 | 0.001 | 0.003178 |
| 200 | 200 | 0.001 | 0.001881 |
| 500 | 200 | 0.001 | 0.001606 |
| 600 | 200 | 0.001 | 0.001613 |
| 200 | 50 | 0.005 | 0.004955 |
| 500 | 50 | 0.005 | 0.00526 |
| 600 | 50 | 0.005 | 0.005159 |
| 200 | 100 | 0.005 | 0.002811 |
| 500 | 100 | 0.005 | 0.002504 |
| 600 | 100 | 0.005 | 0.002547 |
| 200 | 200 | 0.005 | 0.00139 |
| 500 | 200 | 0.005 | 0.001195 |
| 600 | 200 | 0.005 | 0.001092 |

מבין הסטים האילו של הפרמטרים שעליהם הרצנו את המודל,השורה המסומנת בצהוב היא זאת שנתנה את התוצאה הטובה ביותר במובן של צמצום השגיאה.

5)מיפוי TSNE לאחר הרצת השכבה הראשונה של הרשת בלבד:



כפי שניתן לראות , כבר השכבה הראשונה מצליחה למיין די טוב את התמונות,כלומר אותן הספרות מיוצגות ע"י וקטורים קרובים במרחב המצומצם לדו-מימד,ניתן לראות זאת בסרטוט בכך שהנקודות יוצאות מן clusters וקרובות אחת לשנייה (קורדינטות הוקטור מיוצגות ע"י הצירים). ניתן גם לראות שהנקודות שחורגות הן ממספרים שדומים בערך אחד לשני , 3 ו 8 למשל . כמו כן,המקרא בצד שמאל למטה של המפה מייצג את התויות(labels) האמיתיות של הדגימות.

אותו מיפוי לפי הdataset המקורי ללא עיבוד:

אנחנו רואים שעדיין יש clusters, מה שרומז שלתמונות של אותו מספר יש מאפיינים משותפים גם מבחינת פיקסלים בלבד! יחד עם זאת, רמת השגיאה של הנקודות גבוהה בהרבה ויש יותר חריגות. השכבה האמצעית של המודל למעשה עזרה בניפוי השגיאות הללו ובמתן תוצאות מדויקות יותר!