חלק 1:

נתבונן בתת קבוצה של הדאטה של SVHN



התמונות שלנו הן תמונות וניצור מערכת לומדת שיודעת לזהות את המספר בתמונה מפורמט זה בלבד.

*ניצור רשת קונבולוציה שתלמד את התמונות*

defaultCNN(

(conv\_layer): Sequential(

(0): Conv2d(3, 32, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1))

(1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track\_running\_stats=True)

(2): ReLU(inplace=True)

(3): Conv2d(32, 64, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1))

(4): ReLU(inplace=True)

(5): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

(6): Conv2d(64, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1))

(7): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track\_running\_stats=True)

(8): ReLU(inplace=True)

(9): Conv2d(128, 128, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1))

(10): ReLU(inplace=True)

(11): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

(12): Dropout2d(p=0.05, inplace=False)

(13): Conv2d(128, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1))

(14): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track\_running\_stats=True)

(15): ReLU(inplace=True)

(16): Conv2d(256, 256, kernel\_size=(3, 3), stride=(1, 1), padding=(1, 1))

(17): ReLU(inplace=True)

(18): MaxPool2d(kernel\_size=2, stride=2, padding=0, dilation=1, ceil\_mode=False)

)

(fc\_layer): Sequential(

(0): Dropout(p=0.1, inplace=False)

(1): Linear(in\_features=4096, out\_features=1024, bias=True)

(2): ReLU(inplace=True)

(3): Linear(in\_features=1024, out\_features=512, bias=True)

(4): ReLU(inplace=True)

(5): Dropout(p=0.1, inplace=False)

(6): Linear(in\_features=512, out\_features=10, bias=True)

)

)

הרשת קונבולוציה מחולקת ל-2 חלקים משמעותיים.

רשת קונבולוציה (CNN):

כוללת 6 מסנני קונבולוציה בגודל עם ריפוד באפסים וצעד בגודל 1, זאת אומרת משמרים את הגודל של האובייקט, אך מגדילה את המימד עומק (שכבות) של התמונה.

*6 שכבות ReLU ששומרות על המימדים ומאפסות את כל הערכים השליליים, שכבה לא לינארית.*

*3 שכבות MaxPool שבוחרות את האיבר הכי דומיננטי בכל ריבוע ומעבירה אלה רק אותו, המסנן מקטינה את גודל האובייקט ושומרת על המימד העומק, פעולה לא לינארית.*

3 שכבות נרמול ששומרות על המימדים.

*שכבת Drop Out כלי אימון להפחתת סיכוי של over fitting*

*לאחר מעבר ברשת הקונבולוציה אנחנו מקבלים 256 אובייקטים בגודל כל האובייקטים הללו משורשרים לווקטור אחד גדול ועובר לחלק השני של הרשת הלמידה.*

*רשת FC:*

*הרשת כוללת 3 שכבות לינאריות, המקטנית את גודל הווקטור.*

*2 שכבות ReLU ששומרות על גודל הווקטור ומאפסות כל איבר קון מאפס.*

*קיימות גם 2 שכבות Drop out להקטנה של הסיכוי להתאמת יתר*

*נראה כי סך הכל קיימות 9 שכבות לומדות*

*נסכום את מספר המשקולות בכל שכבה לומדת*

*נקבל סך הכל 5,259,104 משקולות שצריך ללמוד את הערך שלהן, בהנחה כי כל המשקולת היא מגודל 32bit = 4 Byte*

*נקבל סך הכל כי ההמערכת ממשקל 21MB.*

*לאחר הכנת הארכיטקטורה נתחיל בשלב הלמידה.*

*תחילה נבחר את ההיפר-פרמטרים של המערכת הלומדת.*

*נרצה למזער את הסיכוי של התאמת ייתר אך עדיין למצוא ערך התכנסות סביר ובזמן סביר.*

*לכן נבחר את הפרמטרים הבאים*

*Epochs = 10*

*Batch size = 256*

*Initial learning rate = 0.0001*

*נשתמש בשגיאת Cross entropy שהכי מתאימה לנו כאשר יש מספר רב של קטגוריון מוצא אפשריות.*

*נשתמש באופטימייזר Adam המבוסס על שיטת SGD.*

*תהליך האימון מתנהל כך:*

*בכל Epoch לוקחים את הסט מידע מתוך הTraining set ב-batch בגודל שהוגדר, נעבירים אותו דרך הרשת.*

*מחשבים את השגיאה המזוהה עם הפתרון שמתקבל כנגד הפתרון הידוע וקובעים ערך שגיאה.*

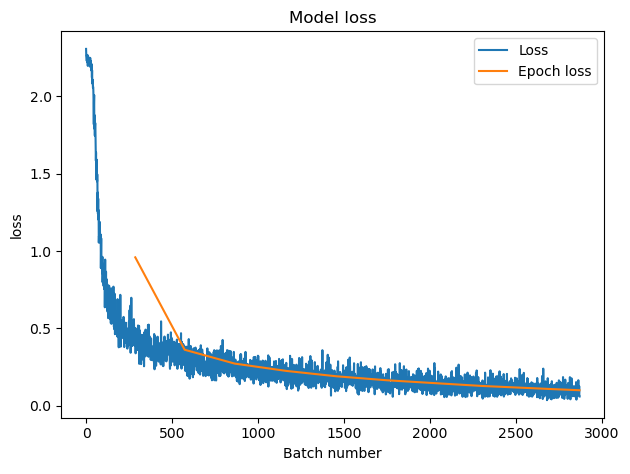
*מתוך השגיאה מחשבים את השגיאה לאחור ע"י חישוב נגזרות בכלל השרשרת, ולבסוף עושים צעד לפי גודל הנגזרת וערך הקצב לימוד.*

*בסוף כל Epoch נבדוק את היכולות חיזוי של המערכת לסט מידע של ערכי האימון והמבחן.*

*לא נשתמש בסט וואלידציה בלימוד בתהליך זה.*

*בנוסף נכפיל את קצב הלימוד פי 0.85 בכדאי לרדת כבסדר גודל אחד כל 10 Epoch ולשפר את ההתכנסות הפונטציאלי של המערכת.*

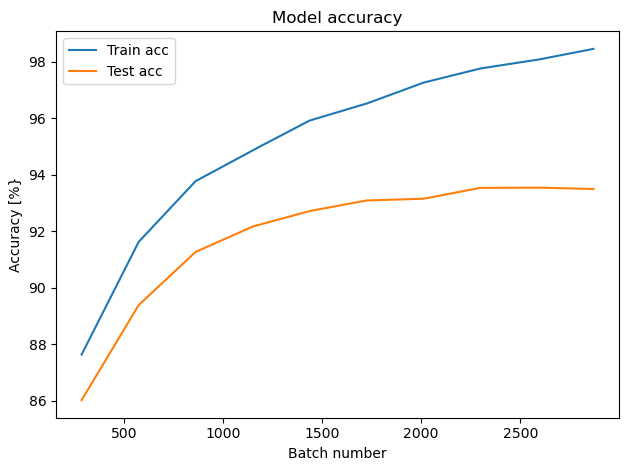
*לאחר הרצה שלמה קיבלנו את הביצועים הבאים:*

* -גרף 1: ערך loss של כל Batch וערך הממוצע לכל epoch*

*נראה כי ההתכנסות ההתחלתית היתה מהירה מאוד ומשם ירדה בקצב, מה שמעיד שהפחתת קצב הלימוד במהלך הלמידה היתה מהלך מועיל, כי נשמר קצב ירידה קבוע.*

*נראה גם כי הגרף פה מציג שאין יתרון ממשי בלשתמש בסט הנתונים הפשוט יותר הנוסף שנתנו לנו.*

*תמונות קלות יותר לא דומות לסט המבחן אז הם לא יאמנו את המערכת היטב בהפוק מתקדם, ואין צורך "לחמם" את הרשת עם מידע קל כי ההתכנסות הראשונית מהירה מאוד. והעלות של מידע רב שלא מועיל יותר מידיי לא שווה את הזמן האימון הארוך יותר שהוא היה גורם.*

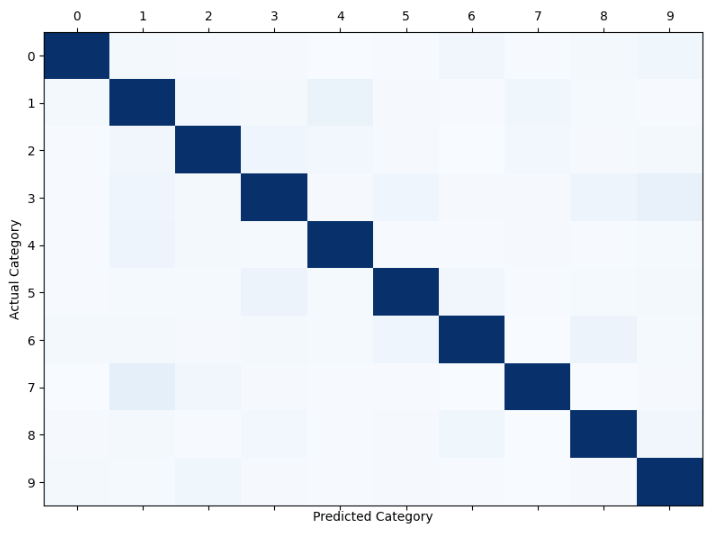
* -גרף 2: ערך דיוק שנבדק כל סוף Epoch על סט אימון וסט מבחן.*

*נראה כי לאורך כל הלמידה הדיוק של המערכת גבוהה יותר לאימון מאשר אל המבחן, מה שהגיוני שיקרה כי המידע שונה מעט ולכן לא מזוהה באותה מידה.*

*כל עוד קצב הלימוד הוא דומה בשני הגרפים נראה זאת כתוצאה טובה אך בשלב מסויים אנחנו מתחילים לראות תופעה קלה של התאמת ייתר לסט האימון, מה שמעיד לכדאיות להגדיל את כמות הDrop out- שקורא במהלך האימון.*

*למרות התאמת היתר בכל זאת דיוק סט המבחן עולה בדיוק שלו עד ה-2 הפוקים האחרונים.*

*ערך דיוק לסט המבחן הסופי שקיבלנו הוא 93.5%*

* -גרף 3: מטריצת החיזוי של המערכת.*

*בצענו חזיו לסט המבחן שהשוונו בין הערך האמיתי והערך שקיבלנו. האיידאל הוא מטריצת היחידה.*

*נראה כי ספרות רבות מתבלבלות עם הערך 1 וכי הספרה 3 נראת למודל שלנו דומה מאוד ל-8,9 ולכן מתבלבל איתו רבות.*