1. נתוני המודל

לצורך בניית המודל השתמשתי במידע משלושה קבצים שונים:

SynthText_val
SynthText

Train

בכל אובייקט מידע יש 5 נתונים:

1) שם התמונה.
2) font - סוג הכתב.
3) Txt - טקסט (תוכן המילה).
4) wordBB – הפיקסלים המרכיבים מילה.
5) charBB – הפיקסלים המרכיבים אות.

כל אחד מהקבצים מכיל כמות שונה של מידע. להלן סיווג המידע לפי קבצים:

נתוני קובץ ראשון SynthText:

'Skylark'=2974, 'Ubuntu Mono'=7334, 'Sweet Puppy'=1930

נתוני קובץ שני train :

'Skylark'=6463, 'Ubuntu Mono'=15164, 'Sweet Puppy'=4443

נתוני קובץ שלישי SynthText_val:

'Skylark'=1915, 'Ubuntu Mono'=4910, 'Sweet Puppy'=1373

ניתן לראות שרוב האותיות הנן מסוג Ubuntu Mono כלומר קיים חוסר איזון בין מספר ההופעות של כל סוג כתב ב "train set".

2. אימון המודל

2.1 Preprocessing

תהליך ה-preprocessing כלל לפני הכל קבלת החלטה לגבי סט נתוני האימון, התלבטתי בין 2 אפשרויות:

(i) להשתמש בכל הנתונים משני הסטים של האימון.
(ii) להפריד סט אחד ולהשתמש בו כטסט לבדיקת טיב המודל.

לאחר בדיקת שתי האפשרויות החלטתי להשתמש בשני הסטים לאימון המודל, בדרך זו התקבלו תוצאות טובות יותר עבור סט הולידציה ששימש אותי לבדיקת טיב המודל. ההימור שקחתי הוא שסט הולידציה ישמש עבורי כטסט ולכן תהיה אינדיקציה קטנה יותר לטיב המודל. בשל העובדה שסט הולידציה מכיל 500 תמונות ו8198 אותיות החלטתי שזאת כמות מספקת לבדיקת טיב המודל.

נתוני אימון המודל - את נתוני האימון הגדרתי כ-x_train ו-y_train:

X_train – אוסף של 38308 פיקסלים המייצגים תמונות של אותיות, נלקח מ - charBB הנמצא בדאטה, במקור שמרתי את האותיות בגודל 400*400 אבל בשל בעיות RAM בגוגל קולאב, הורדתי את כמות הפיקסלים ל 50*50 בעזרת פירמידה גאוסיאנית. הנתונים הם מהצורה [38308,50,50,3] ומסוג asarray. חשוב להדגיש שביצעי נרמול בשלב הכנסת הנתונים לטבלה על מנת שהערכים יהיו בין 0 ל-1.

y-train (labels) – סוג הכתב, נלקח מהמידע txt הנמצא בדאטה ומופיע בצורת STRING, בוצעה המרה של הערכים באופן הבא:

(i) שינוי שמות סוגי הכתב למספרים ואז לוקטורים, הוגדרו שלושה ערכים 0,1,2. מכיוון שהערכים יוזנו למודל שמפרש מספרים לפי סדר אורדינלי (ברגרסיה רגילה/מודל רגיל סוג כתב עם הערך 2 יקבל ערך גדול פי 2 מסוג כתב עם הערך 1, מה שייטה את התוצאות), יש צורך להפוך את ה-LABELS שלנו למשתנה קטגורי (בינארי). ולכן בוצעה ההמרה לוקטורים באופן הבא:
'Skylark'=[1,0,0] , 'Ubuntu Mono'=[0,1,0], 'Sweet Puppy' = [0,0,1].

(ii) השינוי הבא שבוצע על ערכי ה-labels היה הפיכת הטיפוס של הערכים מרשימה למערך.

(iii) השינוי האחרון הינו cast ל-uint8 על מנת לחסוך מקום בזכרון (העברה לערכים בין 0-255).

2.2 בניית המודל

בחרתי ליישם מודל עם פונקציית הפעלה "relu" זאת על מנת למנוע פגיעה בערכי המודל בערכים גבוהים ונמוכים (זאת בניגוד לפונקציות טנגנס וסיגמויד).

סריקת המודל מתבצעת בעזרת חלונות של 3*3 (שינוי החלון לא שיפר את תוצאות).

לצורך טשטוש רעשים הוספתי פילטר מקסימום בגודל 2*2.

הוספתי התאמה (padding) על מנת שהפלט יהיה בצורה של הקלט (וקטור בעל 3 ערכים בין 0 ל1).

בתוך המודל הגדרתי בהתחלה 5 שכבות כאשר הראשונה בגודל 64. לאחר מכן הורדתי את השכבה השניה בגודל 128 ותוצאות המודל השתפרו. לאחר מספר ניסיונות נוספים התוצאה הטובה ביותר הגיעה לאחר הורדת השכבה השניה והשכבה הרביעית (בגודל 256).

מעבר לשכבות הנ"ל הוספתי שכבה נסתרת הכוללת 4096 נוירונים (כ2/3 מ7500 הנתונים שהכנסתי) העלאת כמות הנורונים לאזור ה-7000 הוסיפה מעט מאוד דיוק אך הגדיולה משמעותית את זמן הריצה של המודל.

בשכבה האחרונה בשל העובדה שיש יותר מ-2 קטגוריות השתמשתי ב-softmax שנותן את התוצאה המדוייקת ביותר.

בנוסף לשלושת השכבות ולשכבה הנסתרת, הוספתי פקודת dropout בניסיון להמנע מהתאמת יתר (דבר ששיפר את התוצאות עבור סט הולידציה). וביצעתי שיטוח של התוצאות בשביל להתאים למודל cnn.

2.3 Model compile

בתהליך העיבוד (compile) תחילה בחרתי בפונקצית הפסד במודל cnn עבור יותר מ-2 משתנים, הפונקציה בה בחרתי להשתמש הינה פונקציית categorical crossentropy. בדקתי את מטריצת הדיוק ('accuracy') והאופטמייזר שבחרתי היה adam.
בבחירת האופטימייזר בדקתי את התוצאות עבור עשרת האופטימייזרים הבאים:

- Gradient Descent
- Stochastic Gradient Descent
- Nesterov Accelerated Gradient
- Adagrad
- AdaDelta

- RMSprop
- AdaMax
- Adam
- Nadam
- ftrl

התוצאות הטובות והמהירות ביותר היו של Adam ולכן זהו האופטימייזר בו בחרתי למודל.

קצב הלמידה שהוחלט עליו הוא זה שנבדק ונמצא שנותן את הדיוק הגבוה ביותר.

על מנת למנוע התאמת יתר (overfitting) השתמשתי בתנאי עצירה מוקדמת. הרצתי עבור מאה סבבים (epochs) אבל תנאי העצירה הינו היה וההפסד עבור סט הולידציה גדול מההפסד בסבב הקודם המודל יפסיק לרוץ (סימן שהתחלנו התאמת יתר).

השלב האחרון בהכנת המודל הייתה החזרת הפרמטר restore best weights.

2.4 Model fit

בשל העובדה שבחרתי בשלב ה-processing להשתמש בסט ולידציה כסט בדיקה, ביצעתי validation split ו0.3 מהנתונים שימשו כסט ולידציה. לקיחת חלק קטן יותר מהנתונים לא בטוח יעיד לי על לגיטימיות הואלידציה ולקיחת חלק גדול עשוי לפגוע באימון המודל, לכן החלטתי להשתמש ב30% מנתוני האימון כסט ואלידציה.

מספר הסבבים (epochs) שבחרתי הוא 100 - בשל העובדה שבשלב ה model compile הגדרתי תנאי עצירה עבור מספר הסבבים (epochs), מספר הסבבים שכתבתי בשלב ה-model.fit אינו רלוונטי, בשום הרצה של המודל לא הגעתי קרוב ל100 ולכן זהו מספר גדול מספיק על מנת שלא יפגע בדיוק המודל.

batch size – נבחר 32 על מנת לא לפגוע במהירות המודל (עם מספר קטן) ועדיין להיות אפקטיבי ולהגיע למסקנות מדויקות.

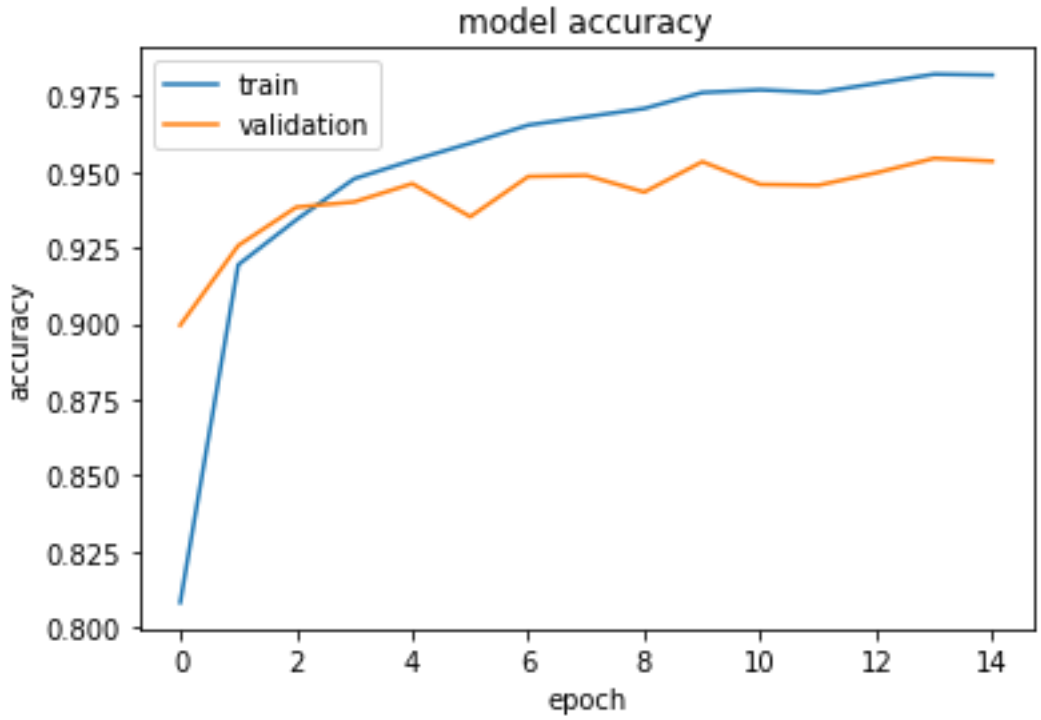
3. <u>הערכת המודל</u>

סט הולידציה שימש אותי כסט הבדיקה, בדומה לנתוני האימון את נתוני הולידציה הגדרתי כ-x_val ו-y_val:

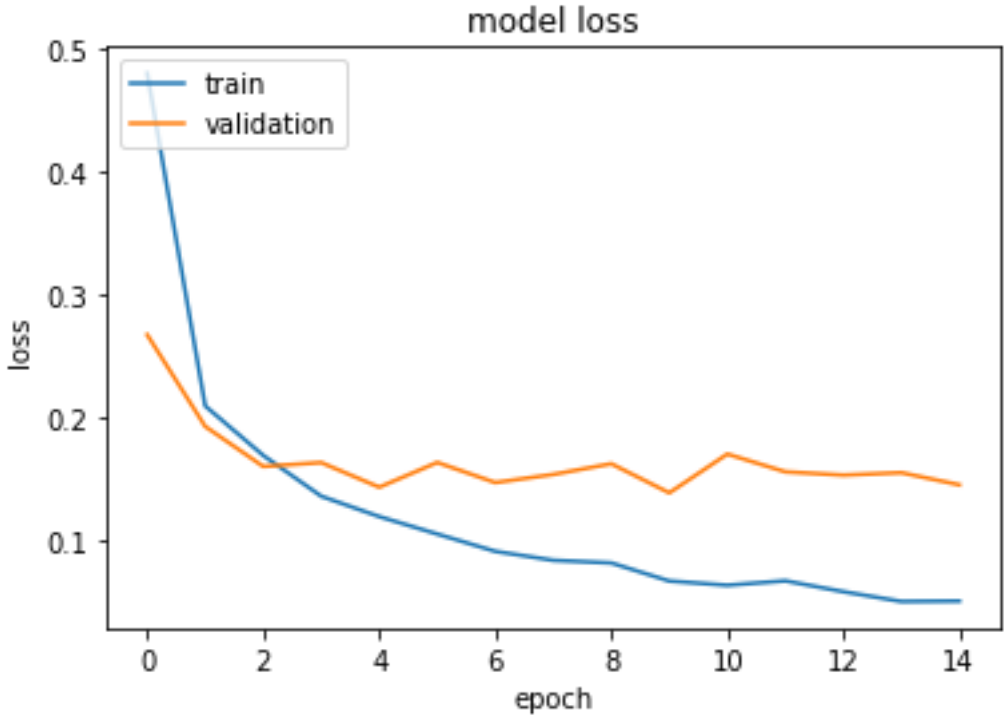
X_val – מכיל 8198 סטים של פיקסלים המייצגים אותיות, נלקח מהמידע charBB הנמצא בדאטה, בדומה למידע ב - x_train גודל התמונה הינו 50*50. הנתונים הם מהצורה [8198,50,50,3] ומסוג asarray.

y-val – סוג הכתב, נלקח מהמידע txt הנמצא בדאטה, בוצעה המרה של הערכים באופן זהה להמרה שבוצעה בסט האימון.

להלן מוצגות דיוק המודל כפונקציה של כמות החזרות (epochs):



ולהלן גרף המציג את loss של המודל כפונקציה קשל כמות החזרות (epochs):



ניתן לראות שזיהוי סוג הכתב עבור שכבת הואלידציה מגיע לשיאו לאחר 9 חזרות כאשר שם הנתונים המתקבלים הינם:

```
Loss: 0.11357492208480835
Accuracy: 96.26%
```

4. חיזוק המודל

בעת חיזוי סוגי הכתב הוספתי עוד תנאי לתוצאות המודל:
בדקתי שאין מילים בטסט באורך 1 או 2. בשל העובדה שאין מילים כאלה וידוע לנו שכל מילה היא באותו סוג כתב הוספתי תנאי שאם יש אות אשר החיזוי לסוג הכתב שלה שונה מהחיזוי של המודל לאות לפניה ושונה מהחיזוי לאות אחריה (בוודאות יש טעות), האלגוריתם יחזיר חיזוי זהה לחיזוי של האות שהייתה לפניה, ולא החיזוי שהמודל זיהה.

דוגמא: עבור האות ה1501 במודל החיזוי הינו לUbuntu Mono' אולם עבור האות ה1500 החיזוי הוא ל Sweet' Puppy וכך גם עבור האות ה1502. במצב זה בוודאות יש טעות, לכן האלגוריתם יחליף החיזוי לאות ה1501 לSweet Puppy' גם הוא.

במידה והאות שלפניה והאות שאחריה זוהו כבעלות אותו סוג כתב ושלושת האותיות נמצאות באותה מילה הדבר יתקן את הטעות בסבירות גבוהה, זאת מכיוון שהסיכוי שהטעות היא בשתי האותיות קטן פי 10 (עבור מודל בעל 90% הצלחה) מהסיכוי שהטעות תהיה רק באות האמצעית.

במידה והאות שלפניה והאות שאחריה זוהו כבעלות אותו סוג כתב והאות האמצעית היא הראשונה או האחרונה במילה פירוש הדבר שייתכן בהסתברות שווה שהטעות היא באות האמצעית או באחת האותיות שלפניה או אחריה. במצב זה שינוי האות האמצעית בהסתברות של 50% יתקן את הטעות ובהסתברות של 50% יצור טעות נוספת. על אף בעיה זאת התוחלת של שינוי זה הינה חיובית (כי מספר האותיות שהן בתחילת מילה או סוף מילה קטן ממספר האותיות באמצע מילה) ולכן בחרתי להשאיר את השינוי הנ"ל.

במידה והאות שלפני והאות שאחרי שונות בסימן, קיים סיכוי זהה לטעות בכל אחת מהאותיות ולכן במצב זה לא נגעתי.

בעיה נוספת שנוצרה לי היא למרות שלקחתי את הערך המקסימלי מבין סיכויי החיזוי של המודל לכל סוג כתב מספר פעמים קיבלתי שהמודל לא ניחש כלל מה הוא הכתב. לצורך כך ושוב בשל העובדה שכל המילים הנן מאותו סוג כתב הוספתי תנאי שאם המודל לא הצליח לחזות את סוג הכתב עבור אות מסויימת הניחוש יהיה זהה לניחוש של האות שהופיעה לפניה (יש סיכוי גבוה מ50% שזה החיזוי הנכון).