מבוא לעיבוד וזיהוי דיבור
67455

האוניברסיטה העברית בירושלים
THE HEBREW UNIVERSITY OF JERUSALEM

Final Project: Introduction to Speech and Audio Processing

מרצה : ד״ר יוסי עדי

מגישים :

דור מסיקה, ת.ז : 318391877

רועי זהבי, ת.ז : 208648154

המודל הסופי

המודל הסופי[1] הינה רשת נורונים המבוססת על RNN עם שכבות קונבולוציה. הרשת אומנה על ידי פונקצית $CTC\ loss$. Evaluation מתרחש באמצעות שימוש ב-Beam Search ומשתמשת במודל שפה $4-gram\ KenLM$.

<u>תוצאות המודל:</u>

	Avg loss	Avg WER	Avg CER
Train	0.3577	0.1344	0.08986
Validation	0.2510	0.069429	0.057056
Test	0.2243	0.0526	0.027317

נתאר כעת את התהליך שביצענו באופן מפורט.

שלב ראשון

התחלנו על ידי מימוש ומדידת התוצאות של מודלים נאיבים בסיסיים, מימשנו 2 מודלים אשר על פי הדוגמה הקרובה להם ביותר מסט האימון.

המודל הראשון עושה זאת באמצעות חישוב מרחק אוקלידי פשוט, והשני מממש אלגוריתם DTW כפי שממשנו בתרגיל 3 בקורס.

ניסינו גם התנהגויות של פיצ'רים שונים והצלחתם במשימת הסיווג תחת מודלים אלו – כאשר ראינו כי התוצאות הטובות ביותר הגיעו עבור שימוש ב-$MFCC$ או $Mel\ Spectrogram$.

להלן ההיפר-פרמטרים בהן השתמשנו כדי להפיק את הפיצ'רים מן קבצי הסאונד:

$$SR = 16000, HOP\_LEN = 160, N\_FFT = 400, N\_MELS = 23, N\_MFCC = 13$$

נציג את התוצאות שקיבלנו מהמודלים הללו:

<u>**Method**</u>	<u>**WER**</u>	<u>**CER**</u>
Euclidean (MFCC)	0.9327	0.7649
Euclidean (Mel)	0.988	0.8232

[1] להלן קישור להורדת משקולות המודל כדי שיתאפשר להריצו עם הקוד שסופק במקביל
https://drive.google.com/file/d/1-914Naz8MyxPLNnzywZ0_WrG7tCnIRzv/view?usp=sharing

DTW (MFCC)	0.9198	0.7170

ניתן לראות באופן ישיר, ולא כ״כ מפתיע כי ה-DTW מפיק תוצאות טובות יותר מאשר מרחק אוקלידי פשוט בין קטעי האודיו השלמים. בנוסף, אלגו׳ זה מפיק ביצועים טובים יותר כאשר אנו משתמשים ב-$MFCC$ מאשר Mel.

כמובן שמדובר בשיטות נאיביות ביותר אשר לא מסוגלות לחזות משפטים שלמים, והצלחתן על מידע שלא נראה עוד בסט האימון מוגבל מאוד. עם זאת, מדובר בקנה מידה בסיסי ובעצם חסם עליון עבור ערכי השגיאה שאנו מסוגלים לקבל עבר יתר הפרויקט.

שלב שני

בשלב השני החלטנו לממש מודל המבוסס רשת נוירונים כדי לחזות. השתמשנו ברשת פשוטה בעלת שכבת RNN ולבסוף שכבה מחוברת לגמרי תוך שימוש באלוגריתם האופטימיזציה $AdamW$, אך באופן די מהיר הרשת החזירה רק את התו הריק והתכנסה לערך הפסד קבוע.

לאחר זמן בו עמדנו במצב זה ומחקר קצר, הבנו כי אימון CTC הינה משימה מורכבת, כאשר החזרת התו הריק נובעת ממינימום מקומי הקיים בפונקציית ההפסד. על כן אנו נדרשים לבסס רשת מעט יותר אקספרסיבית כדי שנוכל לצאת ממינימום זה ולשפר את ביצוענו.

כאשר ניסינו לאמן את הרשת על מספר אפוקים גדול יותר, תוך הגדלת המימד הפנימי של ה-RNN התחלנו לחזות ב-$overfitting$ משמעותי של הרשת, כפי שניתן לראות בפלטים מטה. דבר שגרם לנו להבין כי אנו נדרשים לפונקציה מורכבת יותר כדי שנוכל לתפוס את יתר המידע.

ההיפר פרמטרים שהשתמשו בהם:

$$Hidden\ layers\ =\ 3\quad Hidden\ dim\ =\ 128\quad learning\ rate\ =\ 0.0001$$

ניתן לראות זאת על ידי שגיאת ה-wer, cer בכל אפוק, כאשר ב-70 הראשונים הרשת עוד הייתה תקועה במינימום המקומי והחזירה רק תווים ריקים.

Figure 1 - CER and WER by epoch – basic LTSM

וכן בהפסד על סט הולידציה בכל איפוק שגם כן אינו מתכנס, זאת בניגוד לשגיאה על סט האימון.

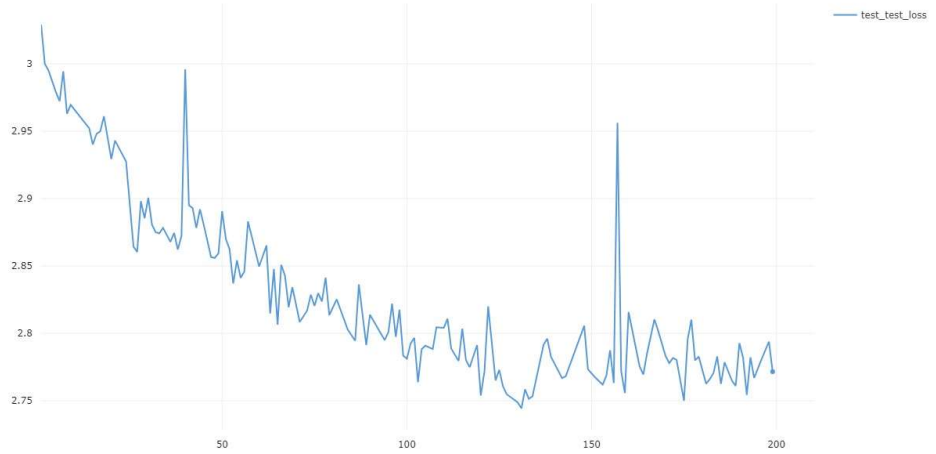

Figure 2 - Validation loss by epoch – basic LTSM

כפי שניתן לראות השגיאות הן גבוהות מאוד, באזור 1~ לכל אחד מן פרמטרי הבדיקה לכן הרגשנו כי אין צורך בתיעודן .

שלב שלישי

בשלב זה ניסינו לבנות רשת מורכבת יותר, הרשת בעלת 2 שכבות קונבולוציה, 3 שכבות $LSTM$ ו-2 שכבות $Fully\ connected\ layer$ בסיומה, עד לקבלת וקטור פרדיקציות בגודל 29.
הפעם בחרנו להשתמש באלגוריתם של RMSProp זאת כדי לנסות להימנע מהתכנסות למינימום מקומי.

המודל לאחר האימון, התחיל לחזות בהצלחה וכן כבר אינו נתקע במינימום המקומי של התווים הריקים, ביצועיו עוד לא היו מספיקים למטרתנו. החלטנו שאין אנו רוצים להמשיך לאמן עבור מספר אפוקים גבוהה, כדי להימנע שוב מ-$overfit$.

המודל עשה שימוש בהיפר-הפרמטרים הבאים :

$$LR = 5e^{-4}\ Dropout = 0.1\ Batchsize = 18\ N_{MELS} = 90\ Epochs = 200$$

הערכנו את ביצועיו תוך כדי אימון על ידי שימוש ב-$Greedy\ decoder$ אשר בוחר בתו בעל ההסתברות הגבוהה ביותר בכל פריים ומבצע קריסה כדי להימנע מכפילויות מיותרות.

להלן התוצאות שהתקבלו עבור מודל זה:

Method	WER	CER
Mid net with CTC RNN layers = 2 Hidden dim = 128	0.8839	0.642597
Mid net with CTC RNN layers = 3 Hidden dim = 256	0.8452	0.6241

להלן הפלטים על אימון הרשת עם הקונפיגורציה השנייה

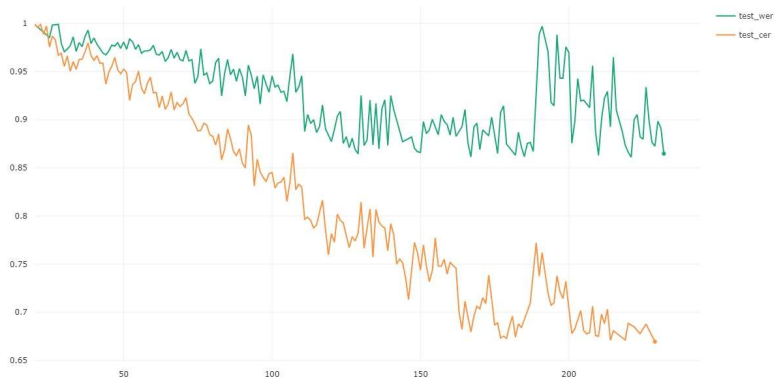

Figure 3 - CER and WER by epoch - Mid NN

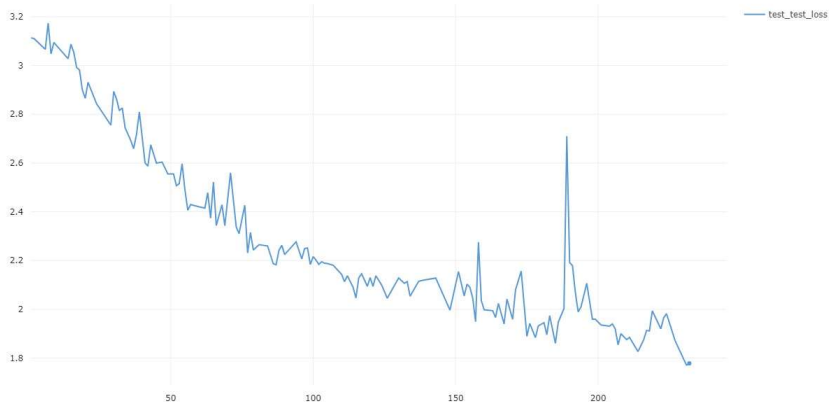

Figure 4 - Validation loss per epoch – Mid NN

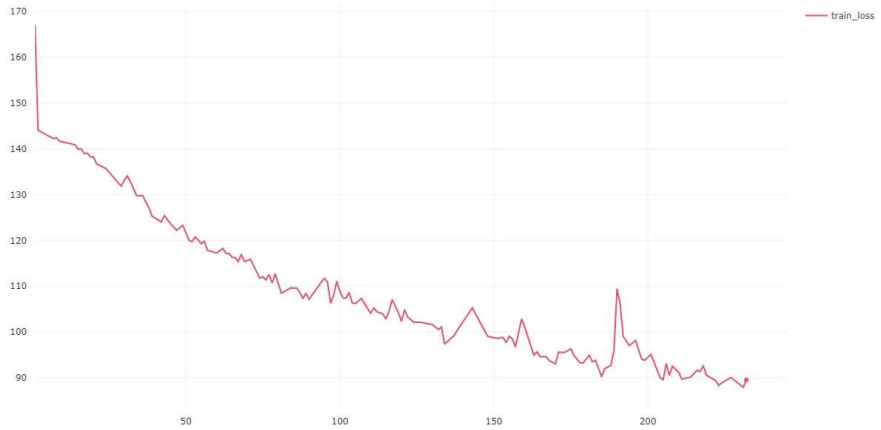

Figure 5 - train loss sum by epoch - Mid NN

שלב רביעי

בשלב זה רצינו לאמן רשת יותר מורכבת, על מנת שתהיה יותר אקספרסיבית ותצליח לתפוס יותר מבנים מורכבים בדאטא, ובכך לקבל תוצאות טובות יותר. לאחר מחקר וחיפוש אחר ארכיטקטורות שונות של רשתות המשמשות ב-ASR החלטנו להתבסס על ארכיטקטורה מנוונת יותר של הרשת $Deep\ Speech\ 2$[2].

באופן מדויק יותר, השתמשנו ברשת המורכבת באופן הבא: שכבת קונבולוציה המגדילה את מספר הצ'אנלים ל-32, בלוק של Residual CNN המורכב משתי שכבות קונבלוציה שביניהן שכבות נרמול ואקטיביציה. לאחריו הוספנו שכבת FC אשר מקטינה את מימד הקלט כך שיתאים למימד הנכנס לשכבת ה-RNN שבאה לאחריו, הממשת Bidirectional GRU עם נורמליזציה ו-Dropout, ולבסוף בלוק קלסיפיקציה אשר מקטין את המימד עד שיתאים למספר המחלקות הדרוש.

הרשת אומנה על ידי שימוש באלגוריתם אדם ($AdamW$) ומתזמן $OneCycleLR$.
להלן ההיפר-פרמטרים:

- $Num_{Mels} = 128$
- $Num_{Epochs} = 100$
- $Batch\ Size = 16$
- $Learning\ rate = 0.0001$
- $Layers_{RNN} = 5\ \ Layers_{CNN} = 3$
- $HiddenDim = 512$
- $Dropout = 0.1$

נציג את התוצאות התקבלות באופן זה על ידי שינויים קטנים שביצענו בהיפר-הפרמטרים:

Method	WER	CER
Final net with CTC RNN layers = 5 Hidden dim = 512 LR = 0.0001	0.1690	0.06953
Final net with CTC RNN layers = 3 Hidden dim = 256 LR = 0.001	0.2156	0.1003

[2] https://nvidia.github.io/OpenSeq2Seq/html/speech-recognition/deepspeech2.html

להלן פלטים שקיבלנו בעת אימון הרשת על 100 אפוקים

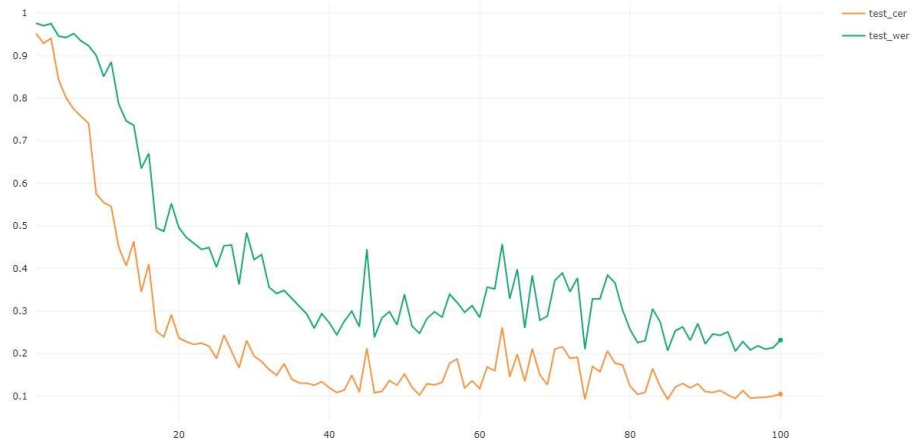

Figure 6 - CER and WER per epoch - Fin net

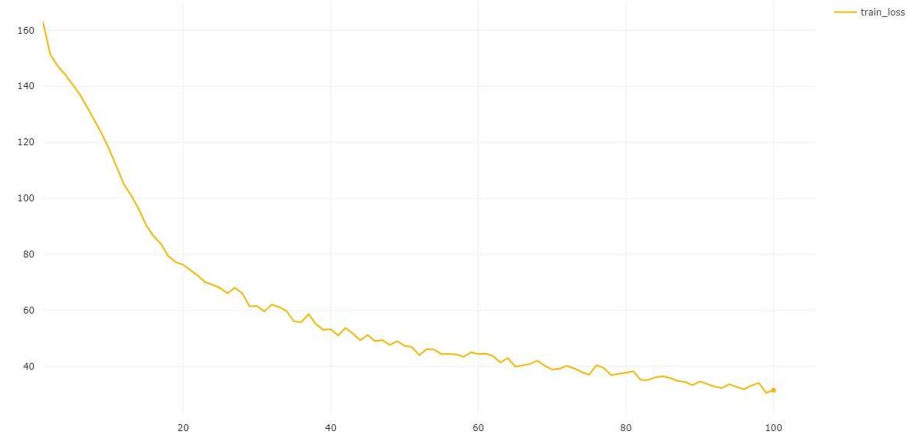

Figure 7 - train loss, sum by epoch - Fin net

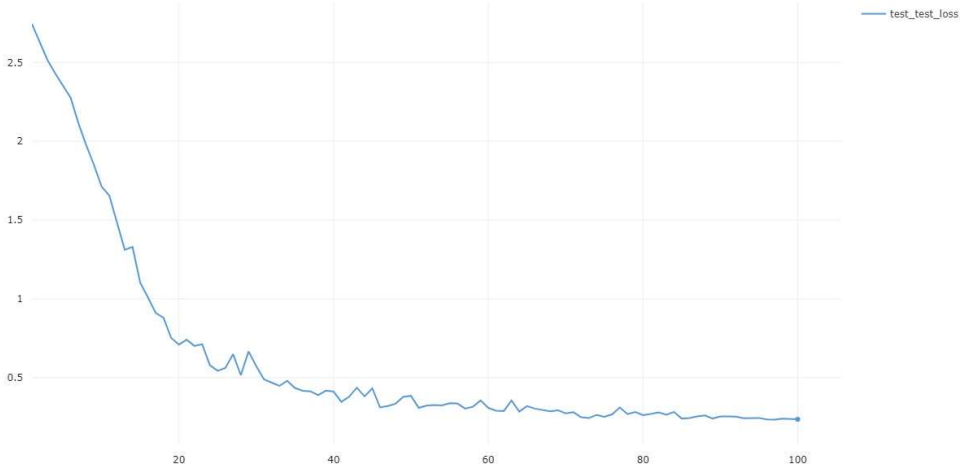

Figure 8 - Validation loss - Fin net

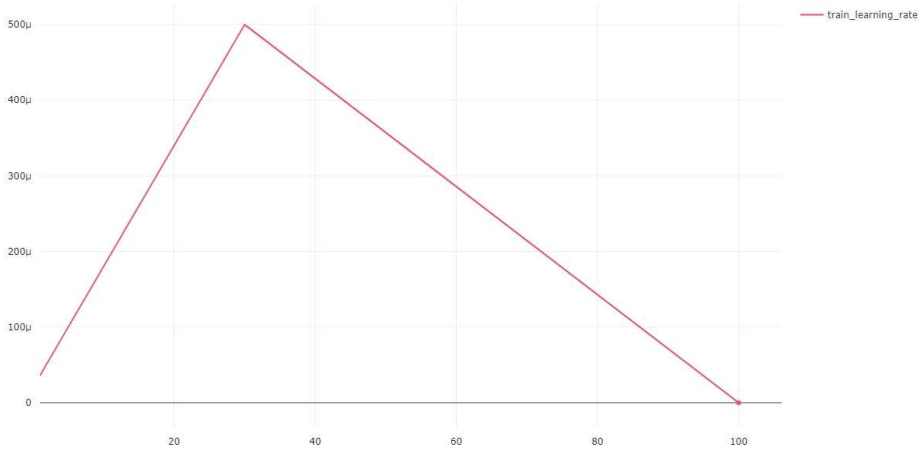

Figure 9 - Learning rate change - Fin net

נציין כי גם בשלב זה השתמשנו ב-Greedy decoder כדי למדוד את השגיאות של הרשת.
מרגע זה הבנו כי בידנו מודל מאומן בעל יכולות טובות והחלטנו לעבור ולשלב אומדנים טובים יותר כדי לשפר את ביצועי המודל מבלי צורך בכוח חישוב שאינו ישיג לנו.

שלב חמישי

כעת ניגשנו לשלב אלגו' Beam Search כפי שראינו בהרצאה, זאת מכיוון שהגישה החמדנית היא סה"כ בהחלט נאיבית.
באופן זה נוכל לעבור על כל המסלולים בעלי ההסתברות הגבוהה ביותר ולאו דווקא נבחר רק את האות בעלת ההסתברות הגבוהה ביותר בכל פריים.
שילבנו למודל קובץ לקסיקון, בהתאם למה שראינו בכיתה ובתיאור העבודה, המכילים את כל המילים שקיבלנו בסט האימון והולידציה, וכן המילון שאיתו אנו עובדים.
להלן תוצאות שקיבלנו על ידי ניסוי עם ערכים שונים באלגוריתם ה-beam search. הערכים אותם מדדנו הם beam size ו word score על הרשת הסופית שהצגנו בשלב 4.

Configuration	WER	CER
$Beam\ size\ =\ 50$ $Word\ score\ =\ 0$	0.2220	0.095695
$Beam\ size\ =\ 50$ $Word\ score\ =\ -0.5$	0.1462	0.060831
$Beam\ size\ =\ 100$ $Word\ score\ =\ -0.5$	0.1458	0.059976

שלב שישי

לבסוף שילבנו מודל שפה כדי לשפר עוד את ביצועי המודל.

השתמשנו במודל שפה $4-gram\ KenLM$ המאומן על הקורפוסים של $Libri\ Speech$, בהם קטעי קריאה באורך כולל של כ-1,000 שעות. מודל שפה זה יאפשר לנו לקבל שיערוכים טובים יותר על ידי מתן יחס להסתברויות שנותן מודל השפה עבור המילה הבאה בכל שלב.

כפי שראינו בשלב הקודם, קיימת רגישות גבוהה גם להיפר-פרמטרים הנובעים משילוב מודל שפה ולהלן התוצאות שקיבלנו בעת ניסוי עם קונפיגורציות שונות.

Method	WER	CER
$Beam\ size\ =\ 50$ $Word\ score\ =\ 0$ $LM\ weight = 2$	0.0712	0.039243
$Beam\ size\ =\ 50$ $Word\ score\ =\ 0$ $LM\ weight = 1$	0.0718	0.034988
$Beam\ size\ =\ 50$ $Word\ score\ =\ -1$ $LM\ weight = 1$	0.0692	0.035069
$Beam\ size\ =\ 100$ $Word\ score\ =\ -1$ $LM\ weight = 1$	0.0526	0.027317

כאשר $beam\ size > 100$ אינו משפיע עוד על התוצאות שהמודל מפיק על סט האימון הנ״ל.

להלן מעט דוגמאות של בדיקת ה-$aligment$ כמול החיזוי של המודל