Automatic Speech

Segmentation

via Praat

מגישים:

מיכאל קליינזיט 323609990

מריה אורצקי 317291037

תוכן עניינים

פרוייקט חלק א:

תאור הפרוייקט 3

תכנית עבודה: דרישות ושלבים לפיתרון 4

הקול האנושי – רקע 5

חלוקת קול אוטומטית – סקירת שיטות 9

סביבת העבודה 12

הרעיון והאלגוריתם 13

הרחבה על רשתות ניורונים 14

תרשימי זרימה 17

מימוש 20

תיכנות הפרוייקט- תאור הסקריפטים 22

סיכונים וקשיים טכניים 24

מסמכי בדיקות 25

מדריך למשתמש 30

הצעות לשיפור 34

פרוייקט חלק ב:

בעיות ופתרונות 35

מדריך למשתמש- מעודכן 36

מסמכי בדיקות 40

מקורות 42

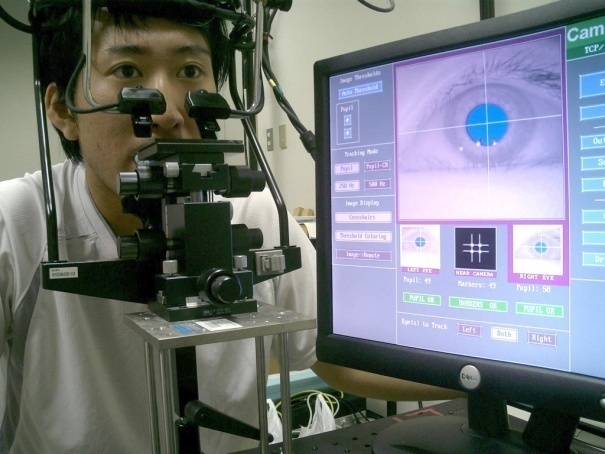
נספחים (בתקייה נפרדת)

תאור הפרוייקט

בחרנו בפרוייקט האמור לשמש בתור כלי עזר בניסוי בתחום לקויות למידה, מנחה הפרוייקט עורך מחקר בתחום זה.   
מטרת המחקר של המנחה הנה להבחין בהבדלים, אם קיימים, בקריאה של אנשים בעלי לקויות למידה לבין קריאה של אנשים ללא לקויות למידה.

כיום ידוע כי כאשר האדם קורא דבר מה בקול רם, הוא קורא קודם עם העיניים חלקים מן המשפט (קריאה שקטה) ורק לאחר מכן הוא אומר בקול רם את מה שקרא.   
לכן אנו מגלים הפרש זמנים בין הקריאה בקול רם לבין ה"קריאה עם העיניים".

המיכשור שרושם את תנועות העיניים



המחקר מתיימר לבדוק את ההבדלים בהפרש זמנים זה, האם שונה, ארוך או קצר?  
המחקר נערך בעזרת הניסוי הבא- הנבדקים מתבקשים לקרוא מהמסך בקול רם משפטים המוצגים להם, תוך כדי הקריאה נרשמות תנועות העיניים שלהם בעזרת מיכשור מתקדם, וכך ניתן להתאים בין הקריאה לתנועות העיניים ולמצוא את הפרש הזמנים.  
  
התוצאות המתקבלות מהניסוי הן- קבצי קול WAV של המשפטים המוקראים, ורישומי תנועות העיניים של הנבדק.  
החוקר מעוניין לדעת איזה חלק של המשפט "קראו" העיניים של הנבדק בכל מילה ומילה שנאמרה בקול רם ועל כן החוקר צריך לדעת את החלוקה של המילים בתוך המשפט, זאת אומרת- היכן מתחילה כל מילה בקובץ הקול. על החוקר לבצע חלוקה של קבצי הקול למילים.  
על מנת להגיע לתוצאות ומסקנות מעשיות מהניסוי הנ"ל על החוקר להעביר את הניסוי לכמות מכובדת של נבדקים, על מספר לא קטן של משפטים (במקרה שלנו כ140 משפטים שונים).  
חלוקת המשפטים הינה הכרחית אך גוזלת המון זמן יקר, מעיקה ומעייפת ולכן התקבלה הצעה – לבצע את החלוקה בצורה אוטומטית.

דף הגדרת הפרוייקט- מהתאר מצורף בנספחים.

תכנית עבודה

מטרתינו היא חלוקה ממוחשבת של קבצי קול בהם נאמרים משפטים ממאגר מסוים הנתון מראש.

הדרישות:

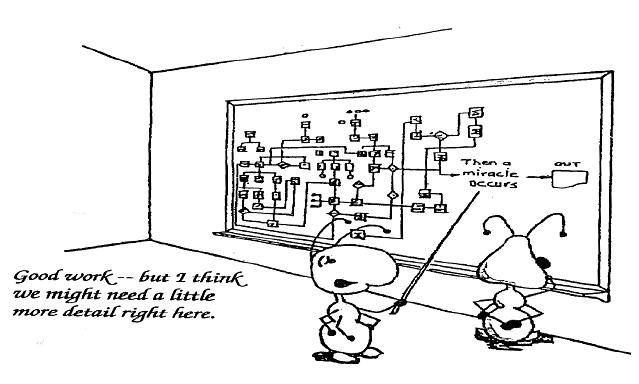
נתבקשנו לכתוב אפליקציה שתחלק את קבצי הקול בצורה אוטומטית, נוחה למשתמש, ותעמוד באילוצים הבאים:

* מספר הנבדקים אינו ידוע
* נבדקים חדשים יתווספו עם הזמן
* הנבדקים הינם משני המינים
* הנבדקים מדברים בטון ועוצמה שונים

בכדי שהמחשב ידע לחלק את המשפטים עלינו להראות לו כיצד לחלק אותם ולנסות ללמד אותו כדי שיבצע את המשימה בצורה אוטומטית.

המשימות לפתרון:

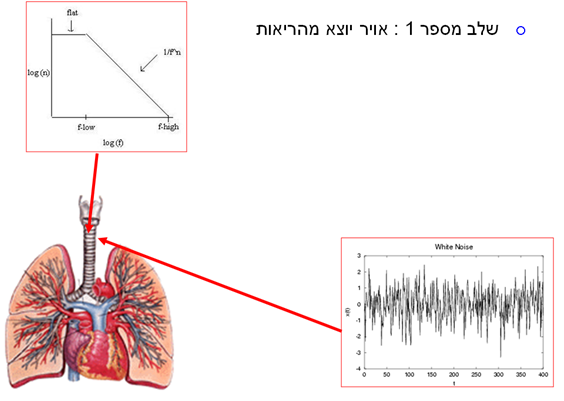
* איסוף מידע על קבצי הקול-   
  ממה הקול מורכב, מהם המאפיינים המיוחדים של כל מרכיב, מה מבין המרכיבים יוכל לעזור לנו במשמתינו,כיצד הקול משתנה במשך הדיבור – הבחנה בין מילים לבין שקט, הבחנה בין מילים שונות, הבחנה בין גבר לאישה בין דיבור מהיר לדיבור איטי וכדומה.
* הכרת סביבת העבודה -   
  אילו כלים קיימים בה וכיצד נוכל להעזר בהם, פונקציות, אפשרויות שונות שנוכל לנצל, אלגוריתמים ממומשים (לדוגמא גילינו כי בתוכנה קיים מימוש נוח מאוד של רשת ניורונים שבה נוכל להשתמש)
* מציאת אלגוריתם לפתרון הבעיה –   
  מציאה בנייה ומימוש של אלגוריתם שיעזור לנו לחלק משפטים בצורה הטובה והמדוייקת ביותר.
* תכנות וכתיבת האפליקציה בסביבת העבודה
* בדיקות תקינות ושיפורים שונים ליעול הקוד



הקול האנושי

הקול האנושי הינו גל, גל הוא הפרעה מכנית בחומר.  
גל הקול מתבטא בשינויי לחץ המתפשטים בחומר. לפי הגדרה אחרת הקול הוא אוסף של דחיסות ודילול מכניים, או גלי אורך שמתפשטים, דרך תווך הניתן לדחיסה (כל חומר למעט ריק). מייצר הקול נקרא גם מקור הגל. גלי קול ניתנים לתיאור בצורה גרפית, כאשר ציר אחד הוא ציר הזמן או המרחב, והציר השני הוא צפיפות האוויר. ההגדרה לקול מתנה את טווח התדר בתחום הנשמע ובעוצמה מעל לסף השמיעה.

הקול נוצר ב-3 שלבים:



# 

מאפייני אות הדיבור:

הדיבור מורכב מסדרת היגויים המרכיבים את ההברות המילים והמשפטים, האדם מסוגל להגות כמה עשרות היגויים שונים.   
היגויים אלה ניתן לחלק לשני סוגים עיקריים:  
 היגויים קולים – אה,אי,או,מ,נ,וי,ב,ד,ג,ל,ר

היגויים אי קוליים- ק,פ,ח,ס,ט,ש,ה

האוויר שיוצא מן הריאות מרעיד את מיתרי הקול נוצר גל אקוסטי. זה עובר דרך מערכת הקול – הגרון,חלל הפה,השיניים,השפתיים הלשון וחלל האף.מערכת הקול מעצבת את גל הקול בהתאם להיגוי אותו מבטאים מדגישה תדירויות מסוימות ומפחיתה תדירויות אחרות. כתוצאה מכך נוצר כל היגוי. גל קול שהספקטרום שלו אופייני להיגוי עצמו ולדובר.  
תדירות הרעידה של מיתרי הקול היוצרים את הגל האקוסטי )ה-Pitch ) אינה אחידה והיא תלוייה בדובר(גבר,אישה,ילד) בהיגוי עצמו ובאינטונציה.   
תחום התדירויות האופייני לגברים הינו 100-150 hz, ולנשים הינו hz 200-300.



היגוי אי קולי בזמן

המשותף לכל ההיגוים האי קוליים הוא שמיתרי הקול אינם משתתפים ביצירתם,הם נוצרים ע"י נשיפת אוויר וחסימה מלאה או חלקית במקום כלשהו בתא הקול- למשל במעבר בין השיניים או הלשון. לדחיקת אוויר זו יש אופי הדומה לאופיו של רעש הנשמע בעת מעבר אוויר דרך פתח צר .

להיגויים אי קוליים לא קיימים פורמנטים. היגוי אי קולי בזמן נראה כמו אות אקראי ללא מחזוריות.

המשותף לכל ההיגויים הקוליים הוא שמיתרי הקול משתתפים ביצירתם. הספקטרום האופייני של היגוי קול מכיל בין 2 ל-3 תדירויות מודגשות,הנקראות פורמנטים. היגוי קולי בזמן נראה כמו גל מחזורי שמחזורו הוא ה- Pitch.



היגוי קולי בזמן



חלוקה להברות עפ"י הפורמנטים

Automatic speech segmentation

שיטות שונות להשגת חלוקה אומטמטית נכונה של משפטים מוקראים בקול.   
את כל תהליך חלוקת הדיבור ניתן לחלק לשני שלבים עיקריים:  
1) השלב הראשון - תעתיק פונטי המיוצר בצורה אוטומטית בעזרת מאגר מילוני גדול של  
 "זיהוי דיבור רציף" (LVCSR)   
2)השלב השני -העברת האינפורמציה הפונטית וקובץ הקול כקלט למערכת אוטומטית  
 שתחלק את הקול.

נשווה בין שתי השיטות,האלגוריתמים העיקריים Viterbi ו- Forward-Backward .

נחקור את התפתחותה של מערכת המחלקת דיבור בצורה אוטומטית בצורה הנכונה ביותר.  
שיטות מסויימות מבוססת על סימנים ומאפיינים אקוסטיים ספציפים או על הבדלים בין סוגי פונמות.רוב מערכות החלוקה האוטומטית מקבלות כקלט את קובץ הקול ואת התעתיק הפונטי שלו.

הסגמנטציה הכוללת בנוייה משני תתי משימות.  
בהתחלה מיוצרים מספר תעתיקים פונטיים אלטרנטיביים על בסיס תעתיק אורטוגרפי נתון, ועל בסיס מזהה דיבור אוטומטי . מזהה הדיבור האוטומטי בוחר את ההתאמות האקוסטיות הטובות ביותר מבחינת ייצוג הפונמות.   
ואז מה שמצאנו משמש כקלט למערכת סגמנטציה המבוססת על אחת מהאלגוריתמים הבאים Viterbi or the Forward-Backward algorithm :

מאורטוגרפיה לפונטיקה

ההמרה האוטומטית מאורטוגרפיה לתעתיקים פונטיים מתבצעת בשני שלבים.   
בהתחלה מספר טכניקות מופעלות בכדי ליצור רשת של קומבינציות אפשריות של הגיים אפשריים. בצעד השני המחרוזת בעלת ההתאמה הפונטית הטובה ביותר נבחרת ע"י אמצעים של מערכת ASR.  
רשת מלאה של תעתיקים פונטיים אלטרנטיביים מיוצרים על בסיס אנפורמציה אורטוגרפית. לא כל יחידה אורטוגרפית היא מילה ברורה . חלקים מסויימים של הדיבור מכילים דיבור אינו ברור ומילים קטועות, היגויי שגויים ואפקטים אחרים של הדיבור.   
טכנולוגיות שונות הומצאו בכדי להתמודד עם הבעיה הנל והתפתחה מערכת הנקראת:  
grapheme-to-phoneme system (g2p) כ fall back-

G2P SYSTEM: מבוססת על Induction Decision Tree (ID3) mechanism

תעתיקים פונטיים מיוצרים מהסוף להתחלה כך שהפונמה המקודדת האחרונה מגיבה לשכן הימני שהוא האינפורמטיבי ביותר.

וזכינו ל6% אחוז שגיאה ברמת המילה.  
מכלי זה נקבל רשת הגיים נוחה.בכדי לבחור את התעתיק הכי מתאים לאות הקול כל האלטרנטיבות הפונטיות נשמרות (המקסימום מועדף) בתוך מסלול האות בתוך מערכת זיהוי הדיבור שלנו וזה בעל ההסתברות הגבוה ביותר מוחזר.   
המודלים של הפונמות מיוצגים ע"י   
three-state left-to-right Hidden Markov Models (HMMs)

Automatic speech segmentation

חלוקת דיבור:

Viterbi vs. Forward-Backward

לאחר שנבחר תעתיק פונטי ניתן להעביר חלוקה אומטית בצורה הבאה: מודלים של משפט מיוצרים ע"י צרוף כל הפונמות הרלוונטיות ואז לכל משפט ניתן לייחס את מידע הדיבור.

Viterbi Segmentation

מחזיר את המסלול האופטימלי במודל בהנתן אות דיבור



כאשר Si הינו צרוף של מצבי HMM (מצב לכל מסגרת זמן), עיקבית עם משפטי המודל S, T הינו מספר מסגרות הזמן .



Automatic speech segmentation

Forward-Backward Segmentation

The Viterbi algorithm נותן לנו רק קירוב של האיכות שאנו מחפשים,אלגוריתם זה מייצר את החסם המגיב ל)1( כאשר החסם האופטימלי לפחות בצורה ריבועית מתאים ל(2) (בגרף הנ"ל) בכדי למצוא את הגבול המוערך הטוב ביותר לפחות בצורה ריבועית ,נרצה לחשב את פונקצית ההסתברות לכל גבול:



במשוואה הנ"ל המשפט S מחוkק בחלק Sl משמאל ובחלק Sr מימין הגבול הרצוי. הפרמטר הנוסף בטא מפצה על ההנחות הלא מדויקות שמנניח ה-HMM -שהניסויים Xi הינם בלתי תלויים.

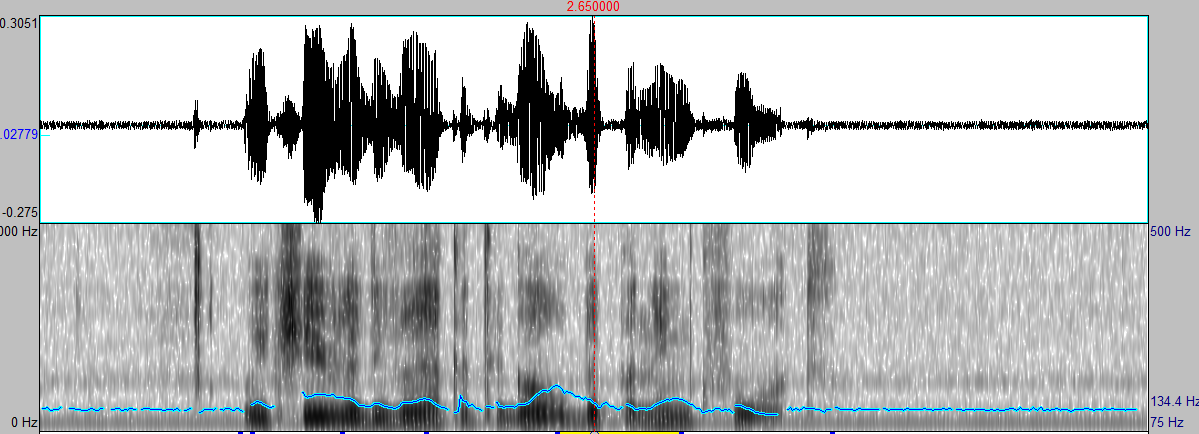
אלגוריתם The Forward-Backwardמאפשר חישוב יעיל של פונקצית ה DENSITY לכל הגבולות במשפט. בהנתן ההסתברות פונקציית הDENSITY של כל גבול היא לפחות הריבוע ששווה:



סביבת העבודה



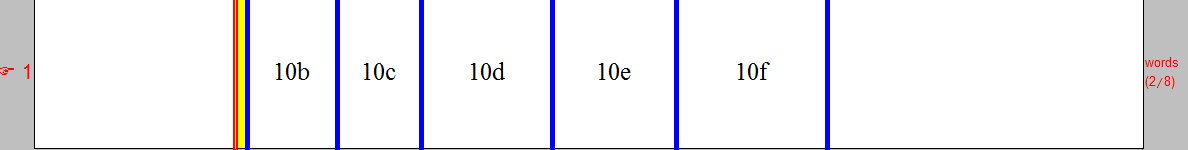
החוקר עורך את הניסוי בסביבת עבודה הנקראת Praat .  
Praat הינה תוכנה המאפשרת לשמוע לעבד ולערוך קבצי קול בצורה קלה ונוחה. התוכנה מאפשרת לאחזר מידע קיים מקובץ קול – Pitch, עוצמה, formants ועוד.  
בנוסף לכך ניתן לראות ספקטרוגרמה של הקול ולבצע בקול שינויים כרצונינו – למתוח, להגביר, להנמיך, לגזור, להדביק וכו'.



תוכנה זו מהווה סביבת עבודה נוחה מאוד גם בשבלנו מכיוון שמכילה בתוכה אפשרות לכתיבת אפליקציות נוספות שיעבדו בשיתוף פעולה עם Praat ולכן בחרנו לעבוד דווקא בה.

קבצי קול בתוכנה מיוצגים ע"י samples, דגימות, דבר המקל עלינו ועוזר לנו להעביר את קובץ הקול הרציף לרשת הניורונים הבדידה שלנו. התוכנה מאפשרת לבצע resample, דגימה מחדש של קובץ הקול כך שמספר הדגימות יהיה גדול או קטן כרצוננו.

החלוקה הידנית של הקול בתוכנה מתבצעת ע"י יצירת קובץ TextGrid שבו ניתן למקם חוצצים במקומות בהם מתחילות המילים ולתת להם שמות.קובץ זה בסופו של דבר נשמר בתור קובץ .txt כפי שנתבקשנו ע"י המנחה.



הרעיון והאלגוריתם

רשת ניורונים היא טכניקה שהתפתחה במדעי המחשב המתבססת על המודל של המוח האנושי.  
המוח שלנו לומד ע"י רשתות עצביות המעבירות מסרים בעזרת אותות חשמליים, ניתן למדל תהליך זה לעולם המחשבים ובעזרתו ללמד מחשב לסווג בין קטגוריות שונות  
אנו נסתמך על השיטה הזאת ונלמד את המחשב לחלק את המשפטים בעזרת רשת ניורונים כמו שהמח עושה זאת.

הרעיון:

נרצה ללמד רשת ניורונים להבחין בין התחלות של מילים, שתגע לקבוע היכן מתחילה כל מילה במשפט כלשהו.  
הרשתות שעליהן נעבור יהיו רשתות Feed Forward המשתמשות ב Back propagation- (הסבר מפורט בנספחים).  
בנוסף להן נקח נתונים סטטיסטיים בכדי ליעל את התהליך.   
רוב בני האדם קוראים משפט מסוים פחות או יותר באותה המהירות, זאת אומרת ישנו טווח זמנים בו אנו יכולים לצפות כי תקרא המילה השניה במשפט X בהסתמך על נבדקים קודמים שקראו את אותו המשפט X.  
לכן נשמור את הזמן הממוצע בו המילה תתחיל- הזמן שבו נצפה לה בפעם הבאה שנתקל באותו המשפט.   
כשנקבל משפט כקלט לסווג האוטומטי נגזור מספר קטעים באזור שבו אנו מצפים לתחילת מילה.  
את הקטעים הללו נעביר לרשת מיוחדת שתדע לסווג משפט זה, והיא תחליט מי מהקטעים הכי מתאים להיות תחילת מילה.

רשתות ניורנים

המח כמכונת חישוב:

המוח האנושי מבצע פעולות חישוביות מגוונות ומסובכות במהירות וביעילות העולות פי

כמה וכמה על אלה של המחשבים המשוכללים ביותר הקיימים כיום, ויש לו תכונות רבות

שהיינו רוצים לראותן במחשבים שלנו: המוח מסוגל לבצע מיגוון עצום של פעולות בו

בזמן, הוא מסוגל ללמוד משימות חדשות ולהסתגל לקליטת גירויים בלתי מוכרים, הוא

מסוגל לעבד מידע מטושטש, רועש, לא עקבי ולא ברור.  
חשבו למשל על נהיגה ביום סגריר, או על קריאת כתב-יד משובש.   
תכונה חשובה של המוח היא עמידותו - יכולתו

החישובית אינה תלויה בפעולתו של תא יחיד זה או אחר. תאים רבים מתים במוח מדי

יום, בלי שנרגיש בפגיעה בתפקודו הכולל. נוסף על כך, המוח הוא בעל נפח קטן יחסית

כ-1500 סמ"ק, וצורך מעט מאוד אנרגיה.

כאשר מתבוננים מבעד למיקרוסקופ בקטע מהמוח, מגלים שהוא מכיל תאי עצב(נוירונים)

רבים מאוד מסוגים שונים. כל נוירון מתקשר עם נוירונים אחרים, והם מעבירים ביניהם

אותות כימיים וחשמליים. במוח כמאה מיליארד נוירונים, ולכל נוירון קשרים עם אלפי

נוירונים אחרים! מספר הנוירונים הרב, רמת הקישוריות הגבוהה והניתנת לשינוי,

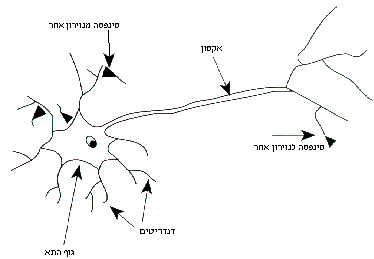
והעובדה שעיבוד המידע מבוצע תוך פעילות של נוירונים רבים בעת ובעונה אחת, ובמידה

רבה של חפיפה בפעולתם - הם המאפשרים למוח לבצע חישובים מורכבים, והם המקנים לו

את התכונות שהזכרנו קודם, כגון העמידות, היכולת לבצע מגוון פעולות, וכושר

הלמידה.

הנוירון הביולוגי:



תא העצב, הנוירון, הוא יחידת העיבוד הבסיסית במוח. בתמונה ( לקוחה מתוך "כמעט 2000" מס' 4,עמ' 21)

גוף התא מחובר ל"עץ" של סיבים, הנקראים דנדריטים, והם קולטים את האותות החשמליים מתאים אחרים, מסכמים ומעבדים אותם. ליד גוף התא מתבצעת ההחלטה הסופית של הנוירון: האם "לירות", כלומר ליצור אות פלט

כתגובה לקלט, או לא. הפלט עובר -בצורת פולס מתח חשמלי- מגוף התא דרך סיב ארוך,

הקרוי אקסון. האקסון מתפצל בקצהו לשלוחות רבות, שבסוף כל אחת מהן נמצאות סינפסות

שדרכן מתבצע הקישור לעץ הדנדריטי של הנוירון הבא. בדרך זו מקבל הנוירון קלטים,

מעבד אותם, ותוצאות העיבוד משמשות קלט לנוירונים אחרים. הפעולות של מעבר הזרמים

החשמליים וסכימת האותות מתבצעות על-ידי מעבר של יונים אטומים טעונים במטען חשמלי דרך קרומים –ממברנות- העוטפים את התא ואת האקסון.

מנוירון ביולוגי לרשת מלאכותית:

המנגנונים האחראים לפעילות הנוירונים ולקשרים ביניהם הם מורכבים ומסובכים, ואינם

ידועים לנו במלואם. עם זאת, ניתן לחקור את הפעילויות החישוביות המתבצעות במוח גם

מבלי לרדת לפרטי המנגנונים התאיים. ניתן להסתכל על כל נוירון כעל יחידת עיבוד

המבצעת פעולה חישובית פשוטה, ועל המוח כולו כעל רשת של מעבדים פשוטים הקשורים זה

לזה. הסתכלות זו על המוח כעל רשת של יחידות פשוטות המקושרות זו לזו הביאה

להתפתחותו של תחום מחקר הידוע בשם "רשתות נוירונים", תחום העוסק בחקר מערכות

מורכבות בעלות יכולת חישובית גבוהה, המורכבות מאלמנטים פשוטים המקושרים ביניהם.

חוקרי רשתות הנוירונים מתמקדים ביכולת החישובית של אלמנטים דמויי נוירון, ואינם

עוסקים, בדרך כלל, בתכונות הפיסיולוגיות של הנוירון במוח. הם אינם מתעניינים

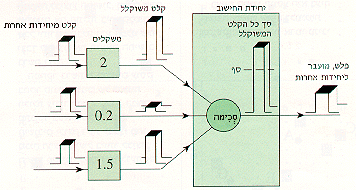
באופן המדויק שבו מתבצע עיבוד הקלטים בדנדריטים ובגוף התא, בריכוזי היונים או

במבנה הממברנות. לגביהם, הנוירון הוא יחידה המקבלת קלט מיחידות אחרות, מבצעת

פעולות מסוימות על הקלט, ומוציאה פלט, המשמש קלט ליחידות נוספות. מקובל להסתכל

על הנוירון כעל יחידה של "סכימה וירייה", כלומר יחידה המבצעת סכימה של קלטים

מיחידות אחרות, ומגיבה ב"ירי" אות, אם סכום הקלטים עובר סף מסוים.



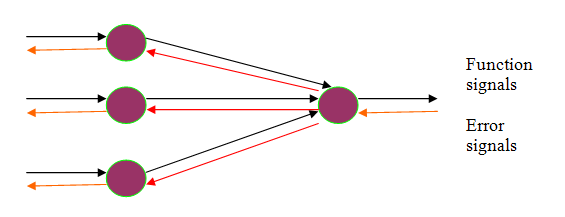
ובכך קובעים את תוצאת החישוב ברשת כולה. בחירת המשקלים היא המפתח, משום שהם מגדירים את התלות בין הקלטים שכל יחידה מקבלת ובין הפלט שלה, כלומר את ערכי המשקלים של כל יחידה לביצוע החישוב. היחידות, מייצגות את המידע ואת תוצאת העיבוד. נוסף על כך, יש לקבוע את הקשרים בין היחידות השונות. לשם כך יש להגדיר מה תפקידה של כל יחידה ברשת ובאיזה אופן ניתן להשתמש בה. כדי לבצע פעולות של עיבוד מידע ולפתור בעיות חישוביות היחידות ברשתות נוירונים מלאכותיות מבצעות פעולות של סכימה משוקללת ותגובה, ומעבירות זו לזו את הפלטים.הפלט, עוצמת התגובה של היחידה, משמש קלט ליחידות והיחידה מגיבה לפי עוצמת הסכום המשוקלל מכל היחידות. כדי להגיב מוכפל הקלט מהיחידות האחרות במשקל המתאים.   
המשקלים משויכים ליחידה ומתארים את מידת ההשפעה של הקלט מכל יחידה על הפלט. עוצמת הקשר אל היחידות השונות מתוארת באמצעות אוסף מספרים, הקרויים "משקלים" .מהיחידות האחרות: יש יחידות שתרומתן לפלט חזקה, ויש שכלל אינן משפיעות על תוצאתניתן לראות שהפלט של יחידה כזאת אינו מושפע במידה שווה מהקלט המגיע אליה.

אלגוריתם Back Propagation

אגוריתם back propagation הוכיח את עצמו בפתרון מספר רב של בעיות כאשר מדובר ברשתות נוירונים בעלות יותר משכבה אחת , אלגורתים זה נקרא:

error back propagation .

העיקרון הוא מעבר על השכבות יותר מפעם אחת (כיוון אחד ואז חזרה).כאשר בדרך "הלוך" כל המשקלים מושמים בצמתים וההשפעה שלהם מתפשטת על פני הרשת כאשר בסופו של דבר מתקבלת תוצאה בשכבה האחרונה ואז במעבר בכיוון ההפוך , המשקלים מתוקנים שכבה אחר שכבה בעזרת חוק תיקון הטעות. וביתר פרוט : התוצאה המתקבלת ביציאה מהרשת מחוסרת מהתוצאה הרצוייה ותוצאת החיסור עוברת נירמול מסויים ומועברת חזרה לרשת בכיוון ההפוך.וכך הלוך ושוב עד שהתוצאה הרצוייה מתקבלת.



**EVALUATION**

**Abstract:**

*All stages adressed as black boxes.*

*Input-output for every stage shown as dashed textboxes.*

*Each specific part will be explained individually.*

**More modules…**

OUTPUT:

Sound **sentence\_num**

**FINAL** TextGrid

Back to learning?

Sound **sentence\_num**

Finer estimate TextGrid

**Trained NN**

**NN module**

Sound **sentence\_num**

Finer estimate TextGrid

**More modules…**

Sound **sentence\_num**

Raw estimate TextGrid

**Mean\_array[x][y]**

**Statistic module**

Sound **sentence\_num**

with the silence cut off.

**insert into praat**

**and cut the silence**

Input : **sentence\_num.wav** file

**TRAINING.**

**Abstract:**

*All stages adressed as black boxes.*

*Input-output for every stage shown as dashed textboxes.*

*Each specific part will be explained individually.*

**INPUT**

Output : Modules (praat objects)

**LEARN**

Individual parsed inputs

for each module

**PARSE**

Praat Objects

Input : files (.wav and .seg)

**TRAINING.**

*All stages adressed as black boxes.*

*Input-output for every stage shown as dashed textboxes.*

*Each specific part will be explained individually.*

Input folder :

* + Folder: subj\_name

= sent\_num.wav

= sent\_num.seg

Praat object pairs:

(Sound sent\_num\_subj\_name,

TextGrid sent\_num\_subj\_name)

**INPUT:**

**PARSE:**

**More modules…**

**STATISTIC**

**NN**

**array[x][y][z] of TextGrid separators**

**x – sentence**

**y – word**

**z - subject**

**Pattern, Categories**

**NN dimensions**

*NN learning parameters?*

**LEARN:**

**. . .**

**Create and Train NN**

**Create mean\_array[x][y]**

Clear ALL data except needed output

**OUTPUT:**

**TRAINED NNMMMMMMM**

**MEAN\_ARRAY[x][y].......**

**. . .**

בעבור כל משפט שנרצה לסווג בצורה אוטמטית נבנה רשת יחודית שתדע לחלק אותו  
כל רשת תראה בצורה הבאה:



כל ניורון בשכבת הפלט שייך למילה מסויימת במשפט, כאשר נרצה לקבוע היכן מתחילה המילה X במשפט מסויים נסתכל על הפלט של ניורון X , הפלט המקסימלי ינתן בעבור הקטע הכי מתאים להיות תחילת מילה X

הניורונים בשכבת הקלט מקבלים קטע קול פוטנציאלי לתחילת מילה, ניורון בודד מקבל דגימה אחת מהקלט. נעבוד עם 1200 נדגימות

נעזר במערך mean array – ישמור את הזמן הממוצע לתחילת כל מילה מהנבדקים שיש לנו עד כה

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 3.4423232 | 3.0000111 | 2.12222344 | 1.8888883 | 1.34999999 | 1.21234599 | 0.63888883 |

בעבור משפט X –  
הזמן הממוצע לתחילת המילה ה-1  
הזמן הממוצע לתחילת המילה ה-2  
הזמן הממוצע לתחילת המיחה ה-3  
.....

הטופולוגיה של הרשת :  
 שכבת קלט: כ1200 ניורונים, ניורון לכל sample של הקול   
שכבה נסתרת :   
שכבת הפלט : מספר הניורונים כמספר המילים במישפט

קלט לרשת:  
 לכל מילה נחתוך קטעים פוטנציאליים מהקובץ הקלט שנתבקשנו לסווג.  
הקטעים הפוטנציאלים הם קטעים בסביבות הזמן הממוצע למילה מהדוגמאות שלמדנו עד כה. הזמן הממוצע נשמר במערך mean array. שכבת הקלט בנויה ממספר הניורונים הנדרש בכדי ליצג קטע קול באורך שבחרנו, ניורון לכל דגימה בPraat, כל ניורון מקבל דגימה אחת מקטע הקול

פלט הרשת:  
 עד כמה קטע מתאים להיות תחילה מילה. בעבור כל מילה נבחן את ניורון הפלט המייצג אותה ונבדוק על איזה מהקטעים שניתן לרשת הניורון הזה מוציא ערך מקסימלי. הקטע שבעבורו התקבל מקסימום בניורון הזה יחשב כקטע הכי קרוב לתחילת המילה

שלב הלימוד:  
1) Training Set - בכדי לבנות רשת עלינו לבנות קבוצת אימון, קבוצת האימון תכלול מספר קטעים באורך 0.3 שניות מתחילת כל מילה במשפט, קטעים אלו ילקחו מקבצי קול של המשפט שאנו מעוניינים ללמוד. הדוגמיות יגזרו מהמשפט בצורה אוטומטית אך החלוקה תהיה ידנית.כל משפט חדש שנרצה להכניס למאגר נצטרך לחלק מספר קבצי קול שלו בצורה  
ידנית ולהכניס לקבוצת הלימוד.

2) ניצור רשת ניורונים בעלת המבנה הבא- שכבת קלט, שכבה נסתרת אחת ושכבת פלט.  
תכנות הרשת בPraat דורש בנית Pattern (קבצי קלט לרשת- דוגמאות של התחלות מילים הידועות מראש) ו- Categories (הסיווג של קבצי הקלט) ,והגדרת המבנה הטופולוגי של הרשת.

3) ניצור מערך ממוצעים –מערך שכל תא בו מכיל את הזמן הממוצע לתחילת מילה,בהתבסס על המערך הזה נוכל לדעת היכן לצפות למילה מסויימת.

שלב הסגמנטציה:  
1) נקרא לתוך סביבת העבודה קובץ הקול של המשפט X, אותו נרצה לחלק בצורה אוטומטית.  
2) נגזור ממשפט זה מספר קטעים בגודל מסויים באזור הממוצע לתחילת מילה– בהסתמך על מערך הממוצעים שיצרנו בשלב הלימוד  
3) נפעיל דגימה מחדש (resample) על מנת שנוכל להתאים את הקטעים החדשים לרשת הניורונים שיצרנו.   
4) נעביר כל אחד מהקטעים הללו ברשת ונבדוק בעבור אלו קטעים התקבל הערך הכי גבוה בניורון בפלט( כל ניורון בפלט מסמן מילה),

5) נקבע את תחילתה של מילה X– ע"י יצירת חוצץ בקובץ הTextGrid ומיקומו בתחילת הקטע בו הניורון של מילה X פלט את הערך המקסימלי.

תיכנות הפרוייקט

האפליקציה בנויה מהסקריפטים הבאים:

Main.praat

שלב הלימוד

שלב הסיווג

Read.praat

Testscript.praatCCCCCCC

Cutsilence.praat DDDDDDD

Detect\_gender.praat

Mean\_arr.praat

MakePattern.praat

תיכנות הפרוייקט

Main.praat– מפעיל את האפליקציה. מקבל מהמשתמש את הפרמטרים ההכרחיים לתחילת הריצה.   
 באפשרות המשתמש ללמד את הרשתות מחדש – שלב הלימוד  
 או לתת לרשת קיימת משפט לסגמנטציה- שלב החלוקה   
 קלט : שלב חלוקה+מיקום המשפט אותו נרצה לחלק או  
 שלב הלימוד+מיקום התקייה המכילה משפטים מחולקים לאימון  
 פלט: אין

שלב הלימוד:   
נרצה לקרוא משפטים מחולקים ולאמן רשתות ניורנים שבעתיד יוכלו לסגמנט משפטים בצורה אוטומטית.  
  
Read.praat- המשתמש מקליד את מיקום התקייה בה נמצאים משפטים מחולקים שיהוו  
 חומר ללימוד הרשתות. הנבדקים השונים צריכים להיות מסודרים בתיקיות –   
 תקייה לכל נבדק, שמה של התקייה יהיה המספר הסידורי של הנבדק   
 לדוגמא: HSCBO001, HSCBO002 וכו'.  
 קלט: מקבלת את הכתובת שהוקלדה בmain-  
 פלט: כל המשפטים שהיו בתוך התיקיות שהיו בתקיה הנתונה בקלט בתור   
 אובייקטים ב-Praat

Mean\_arr.praat- בונה את מערך הממוצעים – היכן בממוצע יתחילו המילים במשפט   
 קלט: קבצי הקול שהם כרגע אובייקטים ב-Praat  
 פלט: מערך ממוצעים הנשמר בתקייה ""script\bin\ , נשמר בתור   
 אובייקט של Praat הנקרא TableOfReal.

MakePattern.praat- בונה את הקלט לרשת את הרשת ומאמן אותה  
 קלט: מערך הממוצעים  
 פלט: רשתות ניורונים מאומנות הנשמרות בתקייה  
 ""script\bin\nn\_female\ או ""script\bin\nn\_male ,בתור   
 קבצי FFNet.  
שלב החלוקה:   
המשתמש יקרא משפט שאותו ירצה לסווג והרשת המתאימה למשפט תחלק אותו.

Testscript.praat-   
מבצע את החלוקה האוטומטית. מפעיל את הסקריפט Detect\_gender.praat כדי לדעת לאיזו רשת להעביר את המשפט- לרשת של בנים או לרשת של בנות.  
לאחר מכן מפעיל את הסקריפט- Cutsilence.praat , המילה הראשונה תתחיל כאשר יגמר השקט בתחילת קובץ הקול, לכן אנו רוצים לדעת מתי נגמר השקט הזה.  
ובסוף נעביר את המשפט ברשת הניורוניום ונקבל קובץ מחולק.  
קלט : המשפט שנרצה לסווג  
פלט: אובייקט seg הנשמר בPraat

סיכונים וקשיים טכניים

במשך העבודה על הפרוייקט צצו מספר קשים ובעיות טכניות שהיה עלינו להגבר עליהן:

סביבת עבודה חדשה:

על מנת לממש את הפרויקט היה עלינו לכתוב בשפת הסקריפט של Praat. לראשונה נתקלנו בתוכנה Praat ולכן היה עלינו ללמוד את שפת הסקריפט בה, אך התגברנו ע"י מדריכים פורומים וסקריפטים מוכנים.

רשת הניורונים:  
לשמחתינו הרבה קיימת רשת ניורונים מובנית בתוך סביבת העבודה, אך השימוש בה אינו טריויאלי. צריך להתאים ולתמרן את הרשת כך שתפתור את הבעיה המסויימת שלנו בצורה הטובה ביותר. לכן עלינו להעביר על הרשת בדיקות רבות ולהבין איך כדאי לבנות אותה על סמך שיטת הניסוי והטעיה.   
בהתחלה לא ברור מה בדיוק לתת לרשת לאבד, מה לתת לרשת בתור קלט. האפשרויות היו- משפט שלם,מילים שלמות או התחלות מילים. כמובן שבחרנו לתת לה את התחלות המילים. כעת עלינו להחליט מהי "התחלת מילה" כמה זמן היא נערכת, ומה מספר הדגימת שיהיה בה.  
כאשר החלטנו והסכמנו על מבנה הקלט בפנינו ניצב האתגר הבא: איך תראה הרשת – כמה שכבות נסתרות כמה ניורונים אמורים להיות בכל שכבה. נתברר כי הביצועים הטובים ביותר נצפו ברשתות בעלות המבנה הבא:

---- ויהיו ----- ניורונים בקלט שכבה נסתרת אחת ומספר הניורונים בפלט יהיה כמספר המילים במשפט..

ההבדל בין גבר לאישה

ידוע כי הקול הנשי נבדל בצורה מהותית מן הקול הגברי. חששנו כי הבדל זה מהווה סיכון לביצועים טובים, חששנו כי רשת ניורונים שתבנה בעבור שני המינים תחלק את המשפט בצורה גרועה מאוד.   
מאוחר יותר כשאר ניסינו לבנות רשת משותפת לשני המינים גילינו כי ההפרדה המתבצעת אינה טובה מספיק.  
לכן החלטנו להקים לכל משפט שתי רשתות –רשת בעבור נבדקות ורשת בעבור נבדקים.  
  
אפשרות להוסיף משפטים  
החוקר ביקש כי נוסיף לתוכנה אפשרות בה הוא יוכל להוסיף משפטים נוספים למאגר, להוסיף אותם גם לקבוצת האימון. בכדי להתגבר על הקושי הזה ניתן לעשות לרשתות למידה מחדש על המשפטים החדשים, החוקר יכניס דוגמאות מחולקות של המשפטים שהוא רוצה לתוך תקיה מסויימת וילחץ על כפתור שיבצע למידה מחדש. וכך יוכל החוקר לחלק בצורה אוטומטית משפטים נוספים שהתווספו

Testing

מטרת שלב הבדיקה הייתה לקבוע את הפרמטרים לתוכנה, איתם היא תעבוד בצורה הטובה ביותר. הפרמטרים שנבדקו הם :

* גודל החלון
* מרחק מהממוצע
* גודל הצעד
* Sampling
* מספר הנוירונים בשכבה הנסתרת

נוהל הבדיקות: הבדיקות נעשו על קבוצת אימון של 3 נבדקות (HSCBO001 HSCBO002 HSCBO005) ונבדקת רביעית (HSCBO006) לבדיקה. (הבדיקות היו רק על נשים בגלל חוסר DATA עבור נבדקים גברים)

על קבוצה האימון אומנו רשתות עם חלונות ב-3 גדלים (0.2,0.3,0.4 שניות), sampling ב-2 גדלים ומספר שונה של נוירונים בשכבה הנסתרת.

לאחר מכן התבצעה חלוקה של משפטים מקבוצת הנבדקות הרביעית, על כל אחת מהרשתות שאומנו, עם מרחקים מהממוצע וגדלי צעד שונים. לכל אחת מהתוצאות (חלוקה של משפט בודד) ניתן ציון מ-1 עד 9 כאשר 9 זה חלוקה מדוייקת ו-1 חלוקה לא נכונה.

תוצאות:

בשלושת הטבלאות הבאות מיוצגות התוצאות של כל הבדיקות שנעשו, כאשר הציון הסופי שניתן לכל בדיקה (לכל חמישיית הפרמטרים) הוא סכום הציונים שניתנו לעשרת המשפטים בקבוצת הבדיקה.

התוצאות במלואן מיוצגות בקובץ EXCEL מצורף – testing.xlsx

Testing

Testing

Testing

Testing

מסקנות: מתוך התוצאות הנ"ל התקבלו מספר מסקנות וכמוכן גם נקבעה חמישיית הפרמטרים האופטימאלית למערכת.

* גודל חלון 0.2 שניות הראה תוצאות גרועות במידה ניכרת מגלי החלון 0.3 ו-0.4
* בעבור SAMPLING וגודל שכבת הנוירונים הניסתרת לא נראה הבדל מהותי בין הגדלים השונים.
* המערכת נוטה לזמן ריצה וכמות זיכרות גדולים בצורה ניכרת עבור SAMPLING גדול, מרחק גדול מהממוצע וגודל צעד קטן. בחלק מהמקרים המערכת אף כשלה בגלל צריכת זיכרון מרובה. במקרים אלו היא קיבלה ציון כולל 0.
* מרחק מהממוצע וגודל הצעד שונים הביאו תוצאות שונות, כאשר ישנה מגמה של תוצאות יותר טובות עבור צעד יותר קטן. כמוכן נראה כי 0.25 שניות זהו מרחק אופטימאלי מהממוצע.

הפרמטרים הסופיים שנבחרו הם:

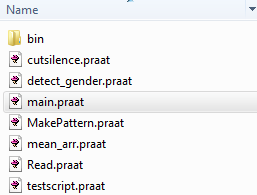
* גודל החלון – 0.3
* מרחק מהממוצע – 0.25
* גודל הצעד – 0.03
* Sampling - 4000
* מספר הנוירונים בשכבה הנסתרת - 400

מדריך למשתמש

התוכנה נכתבה ונבדקה על פלטפורמת PRAAT גרסה 5.1.17. PRAAT היא תוכנת OPEN SOURCE ניתנת להורדה בחנינם מ <http://www.fon.hum.uva.nl/praat>.

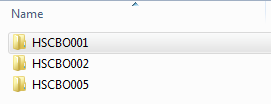
1. התקנה:

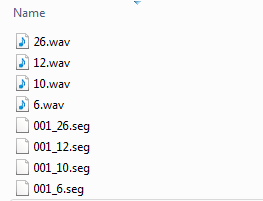
* וודא שמותקנת במחשב תוכנת PRAAT
* פרק את קובץ הZIP בתיקייה הרצוייה.
* וודא שמבנה התיקיות הוא כזה:
  + Testset
  + Trainset\_male
  + Tainset\_female
  + Script
    - קובצי .praat
    - תיקייה בשם BIN שמכילה 2 תיקיות: nn\_male nn\_female.



1. אימון המערכת.

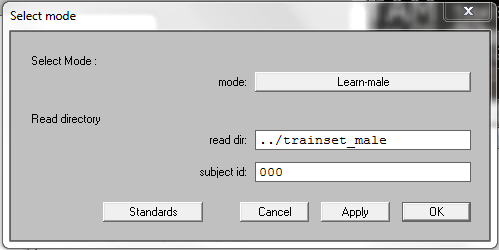
* המערכת מתאמנת על על נבדקים ועל נבדקות בנפרד ולכן יש להכין 2 קבוצות אימון – אחת לגברים ואחת לנשים.
* כל אחת מקבוצות האימון יש לשים בתיקיה נפרדת – Trainset\_male או Trainset\_female בהתאמה ובצורה הבאה:

כל נבדק בתיקייה נפרדת.

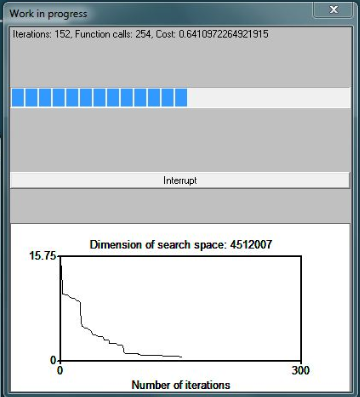


בתוך תיקייה של כל נבדק קבצי .SEG ו- .WAV לכל משפט שרוצים ללמד עליו.

* לאחר מכן יש להריץ את הקובץmain.praat בפראאט. יש לוודא שרשימת האובייקטים בפראאט ריקה לפני ההרצה! מתקבל המסך הבא:

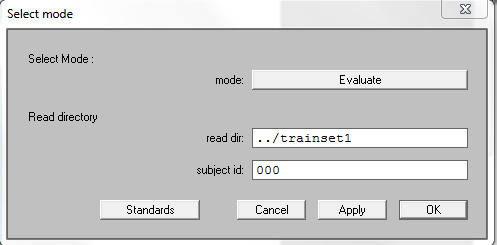


* בשדה mode יש לבחור את האשפרות הרצויה (learn male or learn female ( ובשדה read dir יש לרשום את הכתובת של התיקייה בה נמצאת קבוצת האימון הרצוייה. שדה subject id אינו רלוונטי לשלב האימון. בסיום המילוי הקש OK.
* חשוב: זמן האימון לוקח הרבה זמן ומכביד על המחשב! לא רצוי לעצור את האימון באמצא.



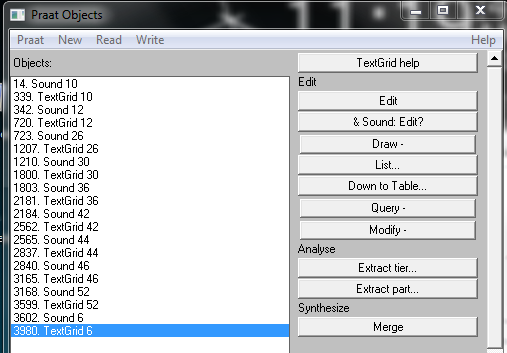
1. שלב ההערכה:

* לאחר שאימנו את המערכת על נבדקים ועל נבדקות ניתן לחלק משפטים של נבדקים חדשים.
* ההרכה מתבצעת על תיקיית קבצי WAV חדשים לא מחולקים.
* שלבי ההערכה:
* הכן תיקייה של קבצי WAV אותם היית רוצה לחלק. ודא שאין קבצים אחרים בתיקייה.
* לאחר מכן יש להריץ את הקובץmain.praat בפראאט.

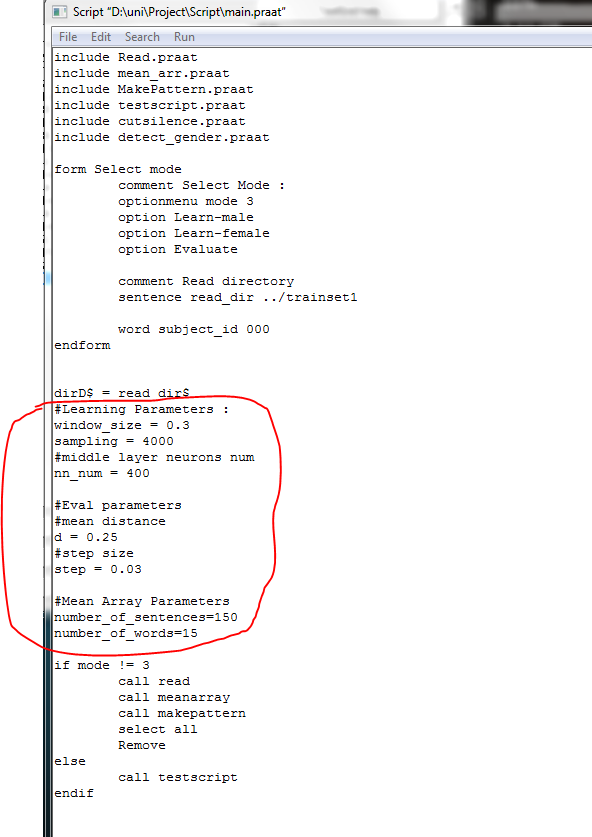


* בשדה mode יש לבחור את האשפרות Evaluate ובשדה read dir יש לרשום את הכתובת של התיקייה הרצוייה. בשדה subject id יש לרשום את מספר הנבדק עבורו מתבצעת החלוקה. בסיום המילוי הקש OK.

בסיום הריצה ניתן לראות את רשימת הזוגות sound – textgrid בפראאט. כמוכן התוצאות נשמרו בתיקייה הנבדקת.



* חשוב: החלוקה לא תמיד מדוייקת! לפעמים יש לבדוק את הקבצים ידנית!

שינוי הגדרות : לאחר בדיקות מרובות נבחרו בתוכנה הגדרות DEFAULT לגדלי רשת הנוירונים, SAMPLING וכו' אך ניתן לשנות הגדרות אלא על ידי עריכת הקובץ main.praat 

מדריך מתכנת לתוכנה-

להמשך פיתוח המוצר

האב טיפוס שבנינו אינו מניב תוצאות מושלמות, ולפעמים דורש תיקון של התוצאות. המערכת שבנינו תוכננה להיות מערכת מרובת מודולות ואף מעודדת תוספות של מודולות חדשות, כפי שניתן לראות בדיאגרמות הזרימה. כרגע ישנן במערכת 2 מודולות – mean\_array ורשתות הנוירונים. האימון של כל מודולה יכול להעשות בנפרד, או שמודולות יכולות להעזר אחת בשנייה. החלוקה נעשת ברצף – כל מודולה נעזרת בתוצאות של המודולה הקודמת או משכללת את התוצאות שלה עם תוצאות קודמות. כל מה שנחוץ להוספת מודולה הוא הוספה של INCLUDE למודולה בקובץ הMAIN וקריאה למודולה מאותו הקובץ.

מספר הצעות למודולות נוספות:

* מודולת רשתות הנוירונים שבנינו מחפשת בתוך הסאונד מילים שלמות. ישנה אפשרות לעשות מודולה דומה שתתיחס לפונמות בודדות במקום למילים שלמות. מודולה כזאות תידרש לפרק כל משפט לפונמות ולחפש כל פונמה בניפרד.
* מודולה שמשתמשת בכלים של ראייה ממוחשבת על מנת למצוא הפרדות בין מילים בספקטרוגרמה של הסאונד המיוצגת בתור תמונת gray scale.
* מודולת רשתות הנוירונים שבנינו מחפשת מילים בתוך קובץ WAV לא מעובד, אך תוכנת הפראאט מספקת כלים מרובים לעיבוד הסאונד שיכולים לעזור בזיהוי של מילים ופונמות, לדוגמא הסתכלות על הpitch או על הformants

פרוייקט חלק ב':

קשיים בפרוייקט חלק א'

בעקבות התהליך "פרוייקט חלק א'" הגענו לתוצאות,התוצאות שהושגו לא מספקות, והדבר יוצר בעיה.   
האפליקציה יודעת לשים מחיצות בין מילים בתוך קובץ קול בכ-50% הצלחה.   
50% הצלחה נמדדו כממוצע בין ההצלחות על כל המשפטים, היו משפטים שבהם האפליקציה צדקה בהרבה יותר מ מ50% (כ-80% הצלחה) ומשפטים שהאפליקציה לא יכלה לזהות בהם את מיקום המילים כלל (כ-20% הצלחה).  
50% הצלחה אינם טובים מספיק, באותה המידה יכלנו לחלק משפט ע"י ניחוש או ע"י סטטיסטיקה בלבד, ללא כל שימוש ברשתות ניורונים.  
היינו יכולים להגיד – אם הממוצע של תחילת מילה x הוא 2.3344566 שניות אז פשוט נקבע שם את תחילת המילה ה-x, או שנגדיר סטייה רנדומית כלשהי מהממוצע ולאחריה או לפניה נקבע את תחילת המילה. באופן זה היינו משיגים גם כ-50% הצלחה.

בעיה נוספת הייתה עניין הדיוק, הדיוק בו האפליקציה ממקמת את החוצץ שמסמן תחילת מילה הוא בטווח גודל הצעד שהוא – 0.03 שניות, היינו רוצים שהחוצץ ימוקם בדיוק רב יותר.

החלטנו לנסות לשפר את האפליקציה כך שהיא תעבוד באחוזי הצלחה גבוהים יותר.

רעיון לפתרון

החלטנו לנסות לשפר את הביצועים של רשת הניורונים שלנו ע"י מתן פחות אנפורמציה. בפרוייקט חלק א' נתנו לרשת הניורונים לעבד את כל האנפורמציה שהייתה בידינו – את קובץ הקול (מקטעים שלו).  
בדרך כלל, ככל שכמות המידע והמורכבות שלו גדלה כך קטנה יכולת הסיווג של רשת ניורונים. לכן החלטנו לתת לרשת פחות מידע, החלטנו לתת לה מאפיין אחד של הקול במקום את כל קובץ הקול ולבדוק האם הפעם האפליקציה עובדת יותר טוב.  
בחנו את מאפייני הקול – pitch ,spectrogram ,pulses ,formants ,intensity חשבנו מה כדאי לתת לרשת,מה יכול להביא אותנו לתוצאה הכי טובה. המאפיינים הללו נותנים לנו מידע כלשהו על הקול, והמידע הזה נראה בצורת נקודות על הספקטרוגרמה, ראינו כי באותו המשפט, לדוברים שונים, צורת הנקודות של ה-intensity על גבי הספקטרוגרמה היא אופיינית וניתן להבחין בעין בין המילים השונות.  
ולכן המאפיין הנבחר, שהחלטנו לתת לרשת הניורונים הוא ה-intensity .  
כעת אנחנו נבצע בדיוק את אותו התהליך כמו ב"פרוייקט חלק א" אבל במקום קובץ קול נטפל ב-intensity.

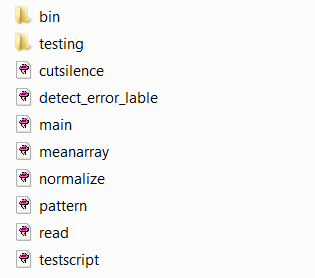
הדיוק- פתרנו את עיניין הדיוק ע"י יעול האלגוריתם.

מדריך (מעודכן) למשתמש

התוכנה נכתבה ונבדקה על פלטפורמת PRAAT גרסה 5.1.17. PRAAT היא תוכנת OPEN SOURCE ניתנת להורדה בחנינם מ <http://www.fon.hum.uva.nl/praat>.

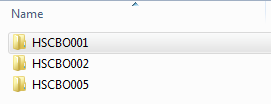
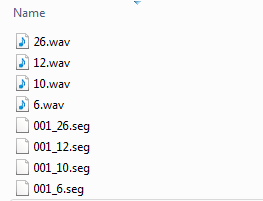
1. התקנה:

* וודא שמותקנת במחשב תוכנת PRAAT
* פרק את קובץ הZIP בתיקייה הרצוייה.
* וודא שמבנה התיקיות הוא כזה:
  + Testset
  + Trainset
  + Script
    - קובצי .praat
    - תיקייה בשם BIN



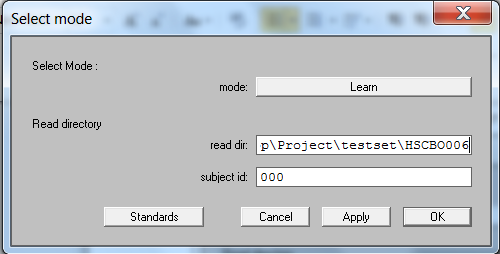
1. אימון המערכת.

* המערכת מתאמנת על על נבדקים ועל נבדקות **ביחד**.
* כל אחת מקבוצות האימון יש לשים בתיקיית TRAINSET

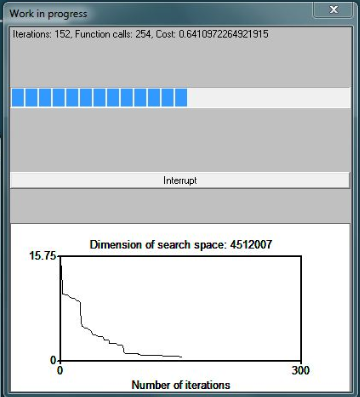
כל נבדק בתיקייה נפרדת.

בתוך תיקייה של כל נבדק קבצי .SEG ו- .WAV לכל משפט שרוצים ללמד עליו.

* לאחר מכן יש להריץ את הקובץmain.praat בפראאט. יש לוודא שרשימת האובייקטים בפראאט ריקה לפני ההרצה! מתקבל המסך הבא:

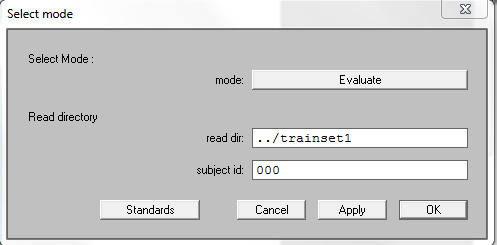


* בשדה mode יש לבחור את האשפרות הרצויה (learn ( ובשדה read dir יש לרשום את הכתובת של התיקייה בה נמצאת קבוצת האימון הרצוייה. שדה subject id אינו רלוונטי לשלב האימון. בסיום המילוי הקש OK.
* חשוב: זמן האימון לוקח זמן ומכביד על המחשב! לא רצוי לעצור את האימון באמצא.



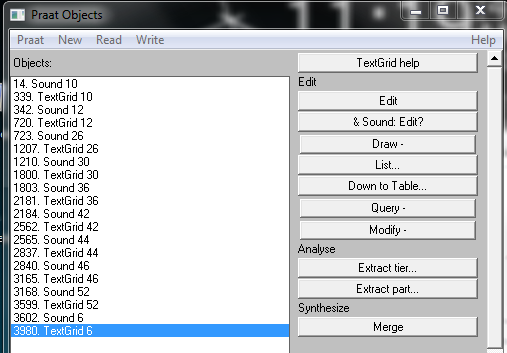
1. שלב ההערכה:

* לאחר שאימנו את המערכת על נבדקים ועל נבדקות ניתן לחלק משפטים של נבדקים חדשים.
* ההרכה מתבצעת על תיקיית קבצי WAV חדשים לא מחולקים.
* שלבי ההערכה:
* הכן תיקייה של קבצי WAV אותם היית רוצה לחלק. ודא שאין קבצים אחרים בתיקייה.
* צור בתוך התיקייה הרצוייה תת-תיקייה בשם grid
* לאחר מכן יש להריץ את הקובץmain.praat בפראאט.



* בשדה mode יש לבחור את האשפרות Evaluate ובשדה read dir יש לרשום את הכתובת של התיקייה הרצוייה. בשדה subject id יש לרשום את מספר הנבדק עבורו מתבצעת החלוקה. בסיום המילוי הקש OK.

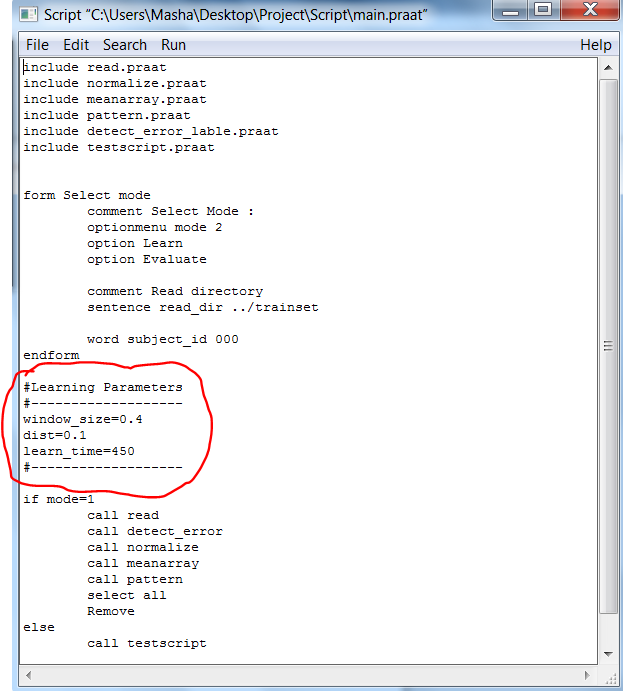
בסיום הריצה ניתן לראות את רשימת הזוגות sound – textgrid בפראאט. כמוכן התוצאות נשמרו בתיקייה הנבדקת.



* חשוב: החלוקה לא תמיד מדוייקת! לפעמים יש לבדוק את הקבצים ידנית!

שינוי הגדרות למידה \ הערכה: יש לשנות את השורות המוקפות באדום (בתוך main.praat)

חשוב: **שינוי דורש לימוד מחדש של הרשת!**



מסמכי בדיקה

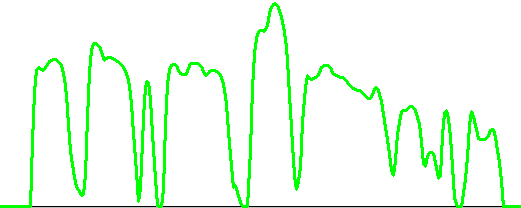
הרצת בדיקות:

בעקבות שינוי האסטרטגיה שלנו, הרשת שלנו קטנה ופשוטה בהרבה ממה שהייתה בפרוייקט חלק א'. והתוצאות שהגענו אליהן טובות בהרבה מהתוצאות הקודמות.   
בדקנו את ביצועי האפליקציה תוך שימוש בערכים שונים לפרמטרים שלנו :

* גודל חלון (window size) : אורך הקטע אותו נתן לרשת
* המרחק מהממוצע (mean distance) : המרחק שהחל ממנו אנחנו מתחילים   
  לקחת דגימות בעבור הרשת
* כמות אטרציות למידה (learn time): כמה אטרציות לימוד נריץ לרשת

תאור גרפי:

הקו הירוק מייצג את ה-intensity, חלקים ממנו ניתנים לרשת בתור קלט.  
הקו האדום הינו החוצץ הממוצע למילה x  
החלק שמסומן בכחול הינו גודל החלון כמה רחוק מהממוצע



מסמכי בדיקה

מסמכי בדיקה

תוצאות:

התוצאות נמדדו באופן הבא: לכל מילה נמדדה הסטיה שלה מהמצב המצופה (הפרש בין המרחקים של תחילת המילה המצויה לבין היכן שהמחשב חשב שהמילה מתחילה),   
למשפט בודד מוצאים את הממוצע של הסטיות הללו, ועושים ממוצע בין כל המשפטים שעברו את הרצת הבדיקה הנוכחית וכך תמדד הטעות.  
ז"א הטעות היא ממוצע הטעות של המשפטים, כאשר הטעות של משפט בודד היא ממוצע הטעויות במילים שבו.

מסקנות:

גודל חלון האופטימלי הוא – 0.4 שניות   
מספר אטרציות הלימוד בטוב ביותר הוא 450  
והמרחק האופטימלי מהממוצע הוא 0.1 שניות

מקורות

מצגת על זיהוי דיבור של רון משה הכט  
[www.math.tau.ac.il/~nin/Courses/NC06/HMMSpeechJun06.ppt](http://www.math.tau.ac.il/~nin/Courses/NC06/HMMSpeechJun06.ppt)

Praat: <http://www.fon.hum.uva.nl/praat>

ויקיפדיה  
<http://en.wikipedia.org/wiki/Wiki>  
  
הספריה החינוכית של מט"ח  
<http://lib.cet.ac.il/pages/frontpage.asp>  
  
אתר הקורס- דחיסת קול ותמונה של אוניברסיטת חיפה  
<http://cs.haifa.ac.il/~nimrod/Compression/>

מצגות של דוקטור דורון בן צבי  
מאמרים:

[www.icis.ntu.edu.sg/scs-ijit/1206/IJIT-1206\_13.pdf](http://www.icis.ntu.edu.sg/scs-ijit/1206/IJIT-1206_13.pdf)  
<http://bero.freqvibez.net/public/segs/icassp03_senseg.pdf>  
<http://www.ele.auckland.ac.nz/~wabd002/Before%202002%20Publications/Improving%20speech%20recognition%20performance%20through%20gender%20separation.pdf>  
<http://appsrv.cse.cuhk.edu.hk/~acm-hk/activity/pg/polyu-nlli.pdf>

חומרים נוספים בנספחים....