תרגיל סיום: Machine Learning

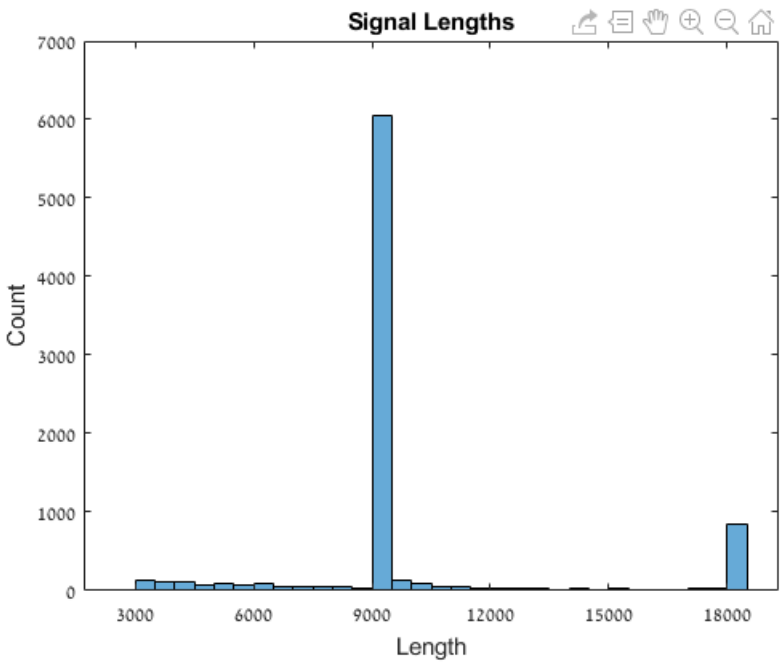
סיווג סיגנלים של ECG

תשובות:

מטרת התרגיל היתה לבנות אלגוריתם (על בסיס אלגוריתם קיים) לסיווג סיגנל (חדש) של ECG (input) לאחת מארבעת הקטגוריות הבאות: (1) נורמלי (N); (2) פרפור עליות (A); (3) הפרעת קצב אחרת (O) או (4) רועש (~). לצורך בניית האלגוריתם, קיבלנו תיקיית Data עם 8528 קבצי mat. שבכל אחד מהם סיגנל של ECG מפציינט בודד (דוגמא לאותות ראה בנספח א'). בנוסף, נתונה טבלת הלייבלים עבור הקבצים הנ"ל: REFERENCE.csv.
לאחר בחינת הסיגנלים הנתונים והפרדת הנתונים לסט של אימון ולסט של ואלידציה, בחרנו להשתמש באלגוריתם המבוסס על RNN מסוג LSTMN וממומש ע"י קוד מאטלב. הקוד המקורי מבצע קלסיפיקציה של נתונים דומים לאלו שקיבלנו, לכדי 2 סוגים בלבד: (1) נורמלי (N) או (2) פרפור עליות (A). בעזרת שיטות שונות לעיבוד אותות ותוך הכנסת שינויים בקוד על מנת שיתאים לסיווג של ארבע קטגוריות (כנדרש), הצלחנו לייצר מכונה לומדת עם דיוק (Accuracy) של 75% עבור סט האימון ושל 65% עבור סט הואלידציה. פירוט כל התהליכים בפסקה זו ינתן בהמשך המסמך.
הקוד מצורף בתיקיית ההגשה, הוראות ההפעלה נמצאות בנספח ב'.

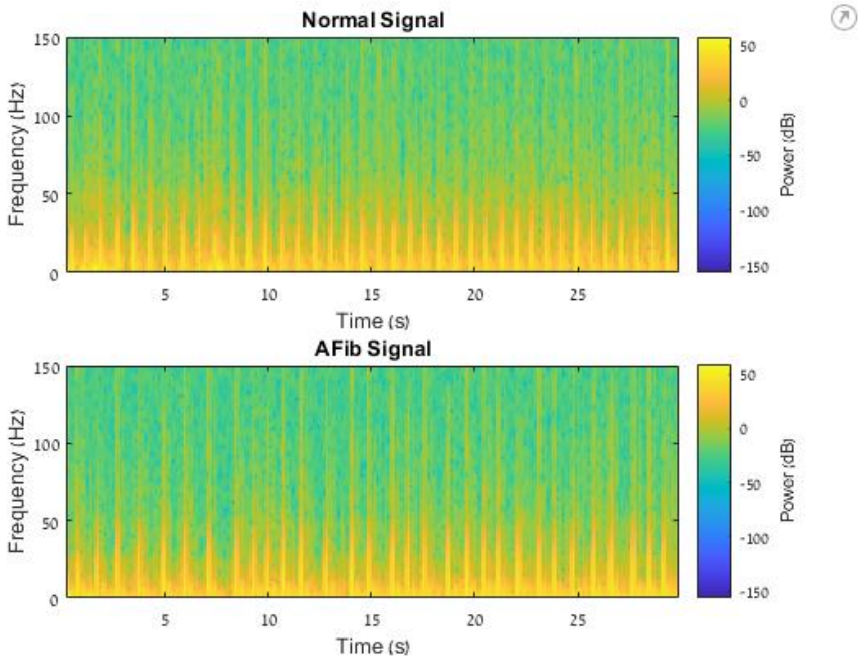
1. **מקור הקוד:** כאמור, לבניית אלגוריתם הסיווג של אותות ה- ECG, נעזרנו בקוד קיים הנמצא באתר הרשמי של MathWorks® כאשר מימוש הקוד הינו ב- MATLAB. העמוד (להלן היפר-קישור לעמוד) מתאר תהליך של בניית רשת תוך שימוש בלמידה עמוקה, לסיווג סיגנלים של ECG לכדי 2 קטגוריות: (1) נורמלי (N) או (2) פרפור עליות (A), תחת הכותרת: ***Classify ECG Signals Using Long Short-Term Memory Networks***.
זו גם הסיבה הראשונה לבחירתנו בקוד זה, כאשר לאחר עיון בקוד, הבנו כיצד נוכל להרחיב את מספר הקטגוריות לסיווג מ- 2 ל- 4 וכן קראנו כי רשת LTSM מתאימה מאוד ללמידת time-series data (כפי שיורחב בתשובה לשאלה 3). סיבה נוספת לבחירת הקוד הינה ה- Data. הנתונים בהם משתמשים בקוד המקורי הם גם 8528 סיגנלים של ECG הלקוחים מהמאגר הבא: *PhysioNet 2017 Challenge*. מדובר למעשה בסט נתונים התואם בתבניתו ל- data שלנו, אך בקוד המקורי מתעלמים מהסיגנלים שהלייבלים שלהם הם (O) ו- (~). לכן, הקוד מותאם לעבודה עם הנתונים שקיבלנו וכך ניתן להשתמש בו בצורה נוחה. הסיבה האחרונה לבחירה בקוד זה, הינה העובדה שמדד ה- Accuracy (פירוט על המדד כפרמטר להערכה ינתן בתשובה לשאלה 5) בבניית הרשת שלהם, עומד על כ- 83.5% (על סט האימון) – מדד מספק יחסית לסיבוכיות (הסבירה) של הקוד.

2. **שיטות עיבוד לנתונים:** בתרגיל זה, ביצענו את תהליך אימון הרשת פעמיים, כאשר בשתי הפעמים כמובן השתמשנו באותה רשת, אך ההבדל הוא ב- input – פעם אחת הפעלנו שיטה בסיסית של עיבוד לנתונים (data) מסוג אחד (בתחום הזמן) ובפעם השנייה מסוג אחר (פיצ'רים בתחום התדר). נציין כי האימון השני הניב תוצאות טובות יותר, על כך נפרט בהמשך. ראשית, עבור 2 האימונים, ביצענו היסטוגרמה על אורכי הסיגנלים הנתונים:



ניתן לראות כי מרבית הסיגנלים באורך 9000 דגימות. במהלך האימון, הרשת מפצלת את ה- data ל- mini-batches ומרפדת/מדללת את הסיגנלים בכל אחד מהבאצ'ים כך שכל הסיגנלים יהיו באורכים זהים. ריפוד/דילול יתר עלול לגרום לאפקט שלילי - הרשת עלולה לפענח סינגל ECG בצורה שגויה, בהתבסס על כמות המידע שנוסף/נגרע ממנו. על מנת למנוע תופעה זו, הקוד מיישם פונקציה המעבדת את הנתונים כך שכל ה- data יהפוך להיות באורך 9000, כאשר סיגנל הקצר מ- 9000 דגימות "יזרק" וסיגנל הארוך מ- 9000 דגימות, יפורק למספר סיגנלים באורך 9000 (והשארית תיזרק).

בפעם הראשונה שהרשת אומנה, השתמשנו בנתונים שפורטו לעיל בתור input. על בסיס הנתונים הללו, ביצועי הרשת היו נמוכים: פחות מ- 50% של Training Accuracy, כ- 23% של Validation Accuracy וזמן אימון (למידה) איטי מאוד. לפיכך, בוצע עיבוד אותות נוסף לנתונים כך שניתן יהיה להשתמש ב- feature extraction (על מנת לשפר את הביצועים) – FFT. בקוד שלנו, השתמשנו בגישה של חישוב time-frequency images (ספקטרוגרמות) ושימוש בהן כ- input לרשת לאימון. לצורך ההסבר, נציג ספטרוגרמה של אחד הסיגנלים המשויך ל- Normal ואחד הסינגלים המשויך ל- AFib (מה- data הנתון לנו):



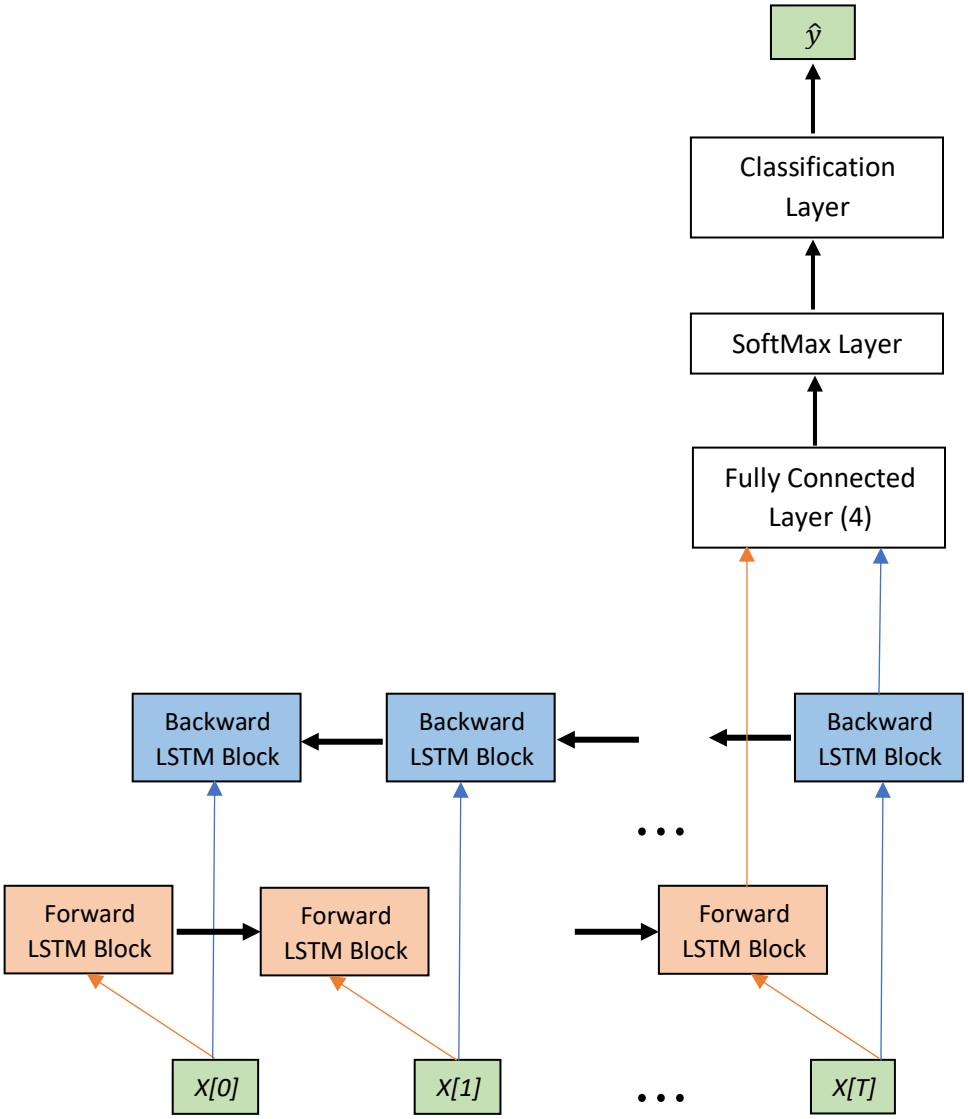
מומנטים של Time-Frequency מחלצים מידע מהספקטרוגרמה כך שכל מומנט יכול להוות סיגנל חד-מימדי. אנו מתעניינים ב- 2 מומנטים:

i. (instfreq) Instantaneous frequency – חישוב ספקטרוגרמה ע"י Fourier Transform על פני 255 חלונות זמן, כך שנקח את מרכזי 255 החלונות.

ii. (pentropy) Spectral entropy – בודק עד כמה spiky flat הספקטרום של הסיגנל (גבוה ברעש לבן, לדוגמא). אופן הפעולה דומה לעיל (התמרת פורייה).

נשלב את הפיצ'רים כך שכעת, כל 'תא' בסט האימון (והואלידציה) יהפוך מסינגל באורך 9000 דגימות למערך דו מימדי באורך 255 דגימות פר מימד (2×255). כעת הקטנו את גודלו של כל תא – זמן אימון קצר יותר בצורה משמעותית וכן הוספנו הסתמכות על פיצ'רים – אימון באיכות גבוהה יותר. לבסוף, בוצעה סטנדרדיזציה (*z-scoring*) על הנתונים שעובדו, כאשר מדובר על דרך פופולרית לשיפור ביצועי הרשת במהלך אימון (פירוט על כך ניתן למצוא בהסברים המלווים את הקוד בעמוד).

3. **המודל:** להלן דיאגרמה של המודל[1]:



ארכיטקטורת מודל ה- RNN: הרשת מורכבת מ- Bi-directional LSTM, לאחריו ישנה שכבת Fully Connected בגודל 4 לסיווג בין ארבעת הקטגוריות (N, A, O או ~) הממשיכה לשכבת SoftMax ולבסוף לשכבת סיווג של אות הכניסה X. X דגום ב- T נקודות כאשר T=9000 עבור האימון על הדאטא סט הראשון ו- $T=2\times 255$ באימון על הדאטא סט אשר עבר עיבוד FT (השני). המוצא $\hat{y}$, הינו סיווג האות לאחת מארבעת הקטגוריות.

המודל (האלגוריתם) אותו מיישם הקוד שלנו לאימון רשת הסיווג הינו RNN מסוג Long Short-Term Memory Network, או בקיצור LSTM. המודל מתאים במיוחד לבעיות קלסיפיקציה וללמידה של time-series data, כפי שקיבלנו (סיגנלי ה- ECG). כל יחידה ברשת ה- LSTM מורכבת מ"תא", אשר זוכר ערכים על פני פרקי זמן שרירותיים ומווסת את זרימת המידע לתא ומחוצה לו. רשת LSTM יכולה ללמוד תלות לטווח הארוך בין

[1] הדיאגרמה מבוססת על מודל המבצע פעולה דומה, מתוך המאמר: Ahmed M., Junye L., Xingliang S. & William W., Classification of 12-Lead ECG Signals with Bi-directional LSTM Network, page 5, URL.

שלבי זמן ברצף. בנוסף, ה- LSTM מכיל חיבורי משוב פנימיים (מבוסס feedback) - שכבת (`lstmLayer`) LSTM יכולה להסתכל על רצף הזמן בכיוון "קדימה", בעוד שכבת LSTM דו-כיוונית (`bilstmLayer`) יכולה להסתכל על רצף הזמן בכיוונים "קדימה" ו"אחורה" כאחד. דוגמה זו משתמשת בשכבת LSTM דו-כיוונית.
לאחר עיבוד הנתונים (signal processing) כפי שפורט בתשובה לשאלה 2, מתבצעת חלוקה של ה- data לסט אימון וסט ואלידציה (ביחס של 90% לאימון ו- 10% לואלידציה, פירוט ינתן בתשובה לשאלה 4). לאחר מכן, נקבעות שכבות המודל:

i. Sequence Input Layer – <u>פרמטרי הכניסה</u>: סיגנלי ECG בעלי מימד אחד באורך 9000 דגימות עבור האימון הראשון ובעלי 2 מימדים (2 frequency features) באורך 255 דגימות עבור האימון השני. מבוטא ע"י X[t] בדיאגרמה הנ"ל.
ii. Bi-LSTM Layer – שכבת הלמידה (דו-כיוונית), כפי שפורט לעיל. בעלת מוצא פנימי של 100 פיצ'רים (ניתן לשליטה ע"י המתכנת).
iii. Fully Connected – גודל 4, מכינה את מוצא הרשת לקלסיפיקציה (4 כמספר הקטגוריות לסיווג).
iv. SoftMax – בדומה לשכבה הנלמדה בהרצאות עבור CNN.
v. שכבת סיווג – <u>מוצא המודל</u>: מחושבת הסתברות השיוך של סיגנל המבוא לכל אחד מארבעת הלייבלים (לאחר הלמידה) והסיווג נעשה על סמך ההסתברות הגבוה ביותר.

בנוסף נקבעים פרמטרי האימון (options) הבאים עבור המודל:

i. מספר אפוקים – עשרה עבור האימון הראשון ו- 30 עבור האימון השני (אפוק=מעבר על כל הדאטא).
ii. גדלי המיני-באצ'ים – בכל נקודת זמן, בחרנו להסתכל על 150 סיגנלים של אימון.
iii. קצב למידה – קבוע על 0.01.
iv. קביעת פרמטרי הואלידציה – בקוד המקורי לא השתמשו בסט ואלידציה, לכן גם לא התקבל גרף של Accuracy ו- Loss עבור הואלידציה. על מנת להשתמש בסט הואלידציה, הוספנו את המידע עליו בפרמטר הואלידציה, וקבענו (דיפולטיבית) כי הערכים יעודכנו על הגרפים כל 50 איטרציות.

4. **חלוקת הנתונים לאימון ולבדיקה:** נציג ראשית את ההתפלגות המספרית של הנתונים ביחס לקטגוריה (לייבל) בה הם נמצאים:

```
summary(Labels)
```

A	771
N	5154
O	2557
~	46

כאמור, תחילה טיפלנו ב- 8528 סיגנלי ה- ECG הנתונים כך שחילקנו אותם לסיגנלים באורכים שווים של 9000 דגימות, כמפורט בתשובה לשאלה 2. להלן ההתפלגות החדשה:

```
summary(Labels)
```

```
A        732
N       4976
O       2668
~         33
```

כלומר כעת התקבלו 8409 סיגנלי ECG. חלוקת הסיגנלים בוצעה באופן הבא: 90% לסט האימון ו- 10% לסט הואלידציה. להלן ההתפלגות המספרית:

Labels	Training	Validation
A	659	73
N	4478	498
O	2401	267
~	30	3

נציין כי:

i. החלוקה מבוצעת בקוד עצמו, כאשר סיגנל מכל קטגוריה מחולק רנדומלית בין סט האימון לסט הואלידציה (כך שהיחס של 9:1 נשמר).

ii. בהמשך, מבוצעת אוגמנטציה על הדאטא, פירוט על כך בתשובה לשאלה 7.

5. **מטריקה להערכת ביצועי הרשת:** על מנת להעריך את ביצועי הרשת (הן על סט האימון והן על סט הואלידציה), השתמשנו במדדים הבאים:

i. Accuracy – מדד לנכונות הקלסיפיקציה.

ii. Loss – מדד לשערוכים (סיווגים) השגויים.

iii. Confusion matrix – ויזואליזציה של ביצועי האלגוריתם, כפי שנלמד בהרצאה.

כאמור, ביצענו שני אימונים של הרשת, הנבדלים בעיבוד והכנת סיגנלי ה- input (אימון וואלידציה). בפעם הראשונה, בוצע עיבוד בסיסי בזמן, כמפורט בתשובה לשאלה 2. לאחר מכן בוצעה אוגמנטציה לסט האימון ולסט הואלידציה, כך שהתקבל מספר שווה של סיגנלים בכל אחת מארבעת הקטגוריות לסיווג (ראה פירוט והתפלגות לאחר אוגמנטציה בתשובה לשאלה 7). בסך הכל, התקבלו 17,886 סיגנלים באורך 9000 דגימות בסט האימון. נציין כי בפועל, לא השתמשנו בכל הסיגנלים הללו משתי סיבות עיקריות: (1) הכמות הגדולה גרמה להאטה קיצונית בקצב הלמידה (הלמידה בוצעה על CPU ולא על GPU); (2) הביצועים של הרשת במדדי ה- accuracy וה- loss הניבו תוצאות טובות פחות (עם overfitting) של סט הבדיקה (אם זאת סט האימון היה טוב יותר בשימוש בכל 17,886 הסיגנלים). לפיכך, בחרנו לצמצם בכפי 2 את מספר הסיגנלים בסט האימון, כך שהגענו להתפלגות הבאה:

```
summary(YTrain)
```

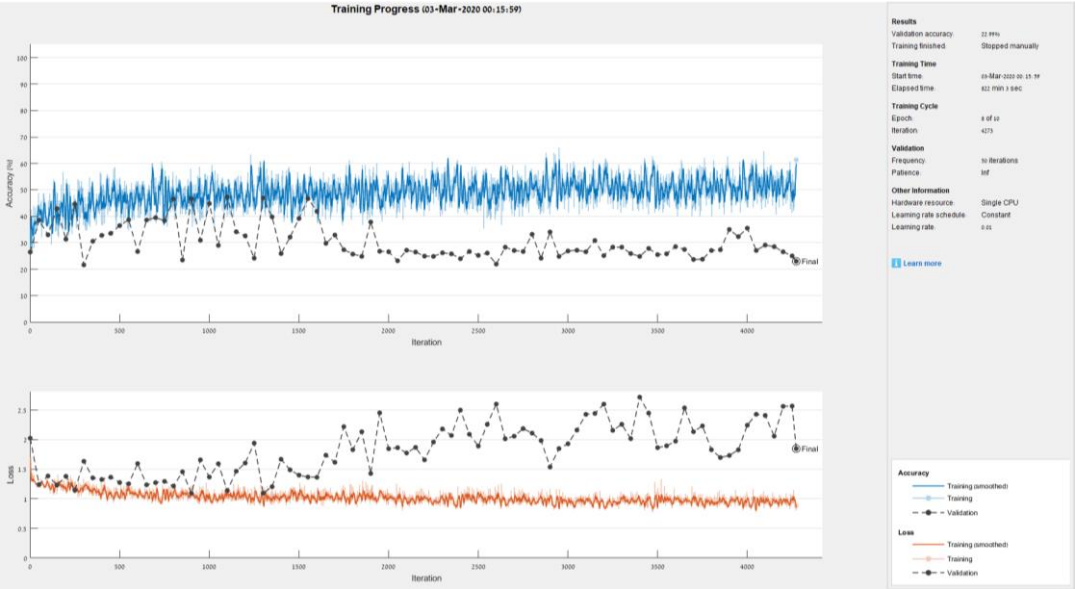
```
A       2556
N       2236
O       2235
~       2250
```

בסט הואלידציה התקבלו בסך הכל 1991 סיגנלים (והשתמשנו בכולם).

<u>בפעם השנייה</u>, בוצע עיבוד בתדר על הנתונים (FFT) על מנת לשפר את ביצועי הרשת באמצעות feature extraction, כמפורט בתשובה לשאלה 2. תהליך האוגמנטציה היה זהה לתהליך שצויין לעיל (בפעם הראשונה) וכן בחירת כמות הסיגנלים לאימון ולבדיקה בפועל נשארה כמקודם. בשונה מהפעם הראשונה, כעת כל תא באינפוט (המייצג סיגנל מקורי) הינו בגודל 2×255 דגימות.

a+b. **ביצועים + גרף השגיאה:**

<u>עבור האימון הראשון:</u>

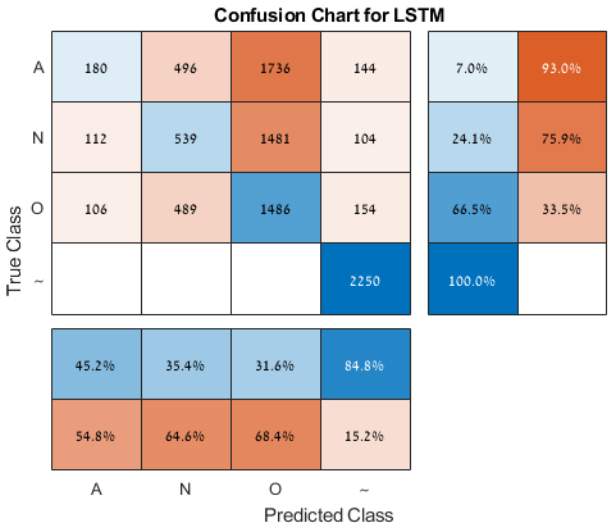


כאשר הפלוט העליון מייצג את ה- accuracy של סט האימון והבדיקה והפלוט התחתון מייצג את ה- loss (גרף השגיאה) שלהם, עפ"י המקרא. הערכים המתקבלים:

Train Accuracy = 48.02%	Valid Accuracy = 22.99%

בנוסף, לאחר 8 אפוקים מתוך עשרה (ביצענו עצירה ידנית של האימון ולא חיכינו עד האפוק העשירי), התקבל Loss גבוה של כ- 2. ניתן לראות מתוך גרף ה- Loss כי עדיף היה לעצור את האימון בסביבות ה- 1500 איטרציות, משום שאז גרף השגיאה של הואלידציה החל לעלות – מה שמעיד על אוברפיט. החלטנו לתת לאימון להמשיך, משום שהביצועים שלו היו נמוכים בכל מקרה – פירוט על כך בתשובה לשאלה 6.

בנוסף קיבלנו את ה- Confusion Matrices הבאות עבור סט האימון:



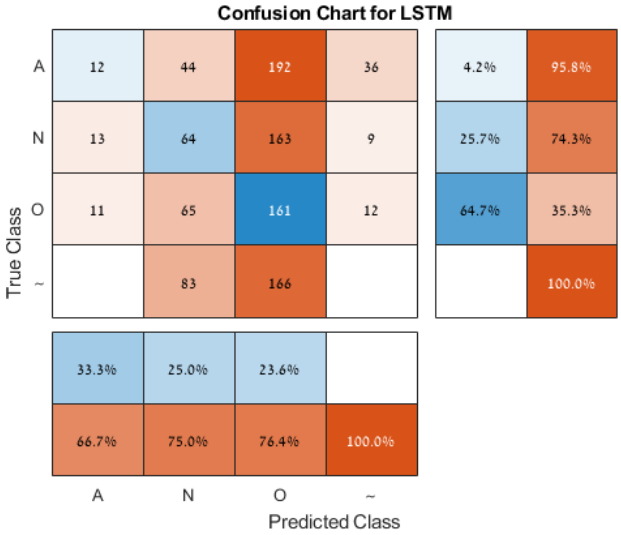
Confusion Matrices הינה שיטה להערכת ביצועים בתחום של ה- machine learning בדגש על בעיות classification שהתרגיל שלנו נמנה על הסוג הזה של בעיות.

כל שורה של המטריצה מייצגת את המופעים ב- class בפועל ואילו כל עמודה מייצגת את המופעים ב-class החזוי. תאים באלכסון מתארים תצפיות שמסווגות נכון ותאים מחוץ לאלכסון תואמים תצפיות מסווגות לא נכון.

הטבלאות באחוזים מצד ימין ולמטה מתארות שתי שיטות להסתכל על התוצאות. הטבלה מצד ימין מראה לנו באחוזים מתוך class מסויים שניתן לנו, כמה המערכת חזתה נכון (העמודה השמאלית) וכמה לא נכון (העמודה הימנית). לדוגמא, המערכת חזתה נכון 66.5% מאותות ה- (O) כ- (O). הטבלה התחתונה מראה לנו באחוזים מתוך class מסויים שהמערכת חזתה וניפקה עבורנו תוצאות, כמה באמת היה נכון וכמה שגוי ביחס לסך הסיגנלים שהמערכת קבעה ששייכים ל- class זה (המספר העליון כמה המערכת צדקה באחוזים והתחתון הוא כמה היא טעתה).

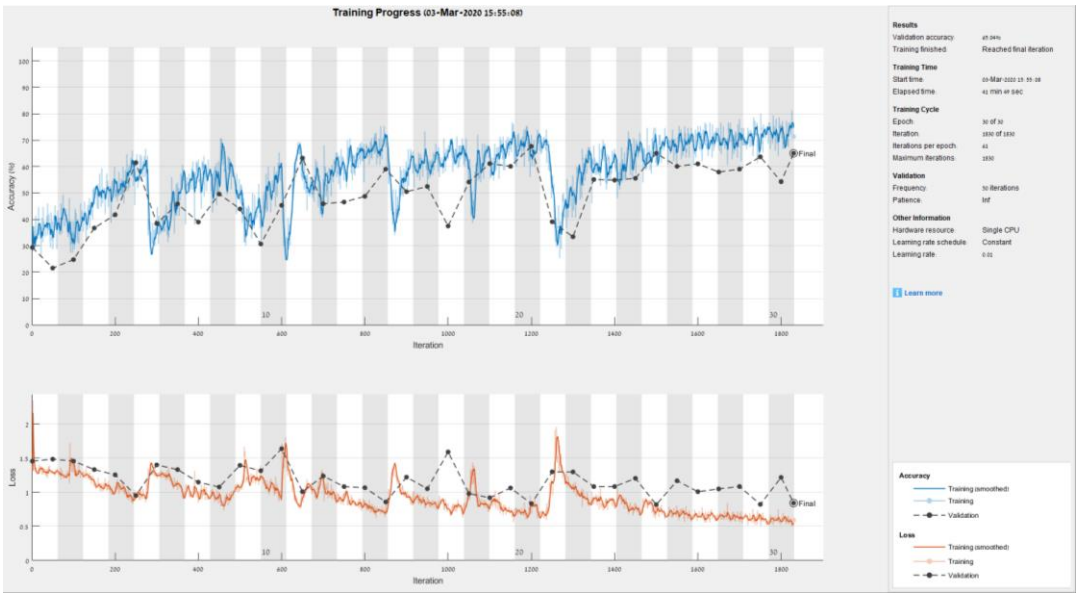
ניתן לראות כאן שהביצועים של הרשת אפילו עבור האימון לא היו מספקים, בהמשך אנחנו מפרטים לגבי הואלידציה.

ועבור סט הואלידציה:



גם כאן הטבלה (מטריצה) היא מדד להערכת הביצועים כפי שהסברנו למעלה, הפעם המטריצה היא עבור סט הואלידציה וגם כאן ניתן לראות (כפי שיכולנו לצפות מהאימון) שהתוצאות אינן טובות והמערכת התקשתה מאוד לאבחן נכון את הסיגנלים.
זו הסיבה שבהמשך לאחר שהמשכנו וביצענו את האימון השני (שהיה תלוי בפ'יצרים) של התדר, וקיבלנו בו תוצאות טובות יותר, החלטנו להמשיך ולעבוד איתו מאשר עם האימון עם הנתונים שמעובדים בשיטה הראשונה (בזמן בלבד).

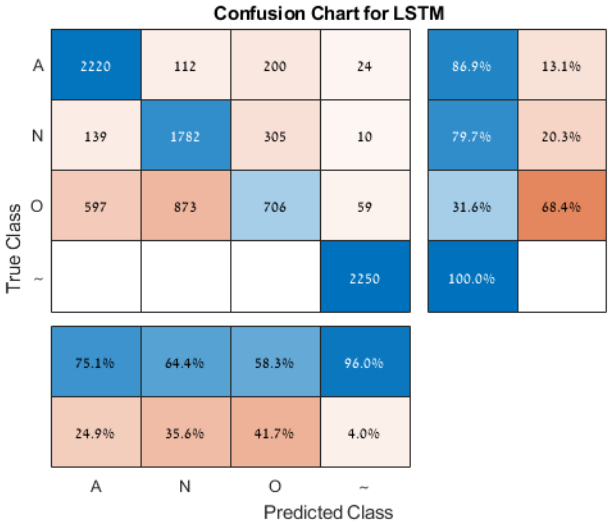
עבור האימון השני:



כאשר הפלוט העליון מייצג את ה- accuracy של סט האימון והבדיקה והפלוט התחתון מייצג את ה- loss (גרף השגיאה) שלהם, עפ"י המקרא. הערכים המתקבלים:

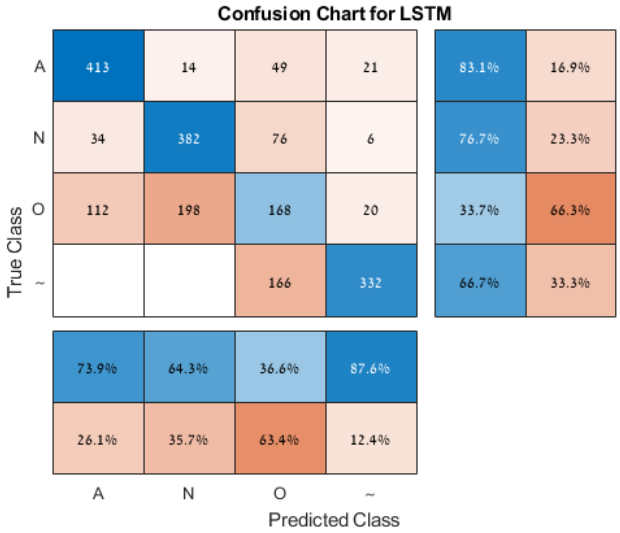
Train Accuracy = 75%	Valid Accuracy = 65.04%

ראשית, ניתן לראות שיפור ניכר לעומת האימון עם הנתונים לפני עיבודם בתחום התדר. בנוסף, הפעם לא עצרנו ידנית את האימון כיוון שהביצועים היו טובים.
נציין בנוסף כי כאשר הוספנו אפוקים, לא חל שיפור כי לאחר מכן גרף ה-Accuracy נהיה 'פלאטו', פירוט על כך בתשובה לשאלה 6.
בנוסף קיבלנו את ה- Confusion Matrices הבאות עבור <u>סט האימון</u>:



כפי שכבר ציינו למעלה מטריצת ה- confusion היא מדד להערכת ביצועי המערכת, באימון זה ניתן כבר לראות תוצאות "יפות" הרבה יותר. ראשית, המספרים באלכסון גדולים ביחס לשאר בשלוש מתוך 4 השורות (מלבד ה- O), מה שמסוכם בטבלה הימנית. שנית, גם בטבלה התחתונה ניתן לראות ביצועים די טובים. ניתן להסיק כאמור, שהביצועים השתפרו, לדוגמא כמעט 87% מסיגנלי ה-AFib סווגו בצורה נכונה. אם זאת, ניתן לראות כי יש "רגישות שלילית" מסויימת לסיגנלי ה- O, כאשר רק 31.6% מהם סווגו בצורה נכונה. דרך לייעול מפורטת בתשובה לשאלה 8.

ועבור סט הואלידציה:



גם כאן בואלידציה, בסה"כ עבור 3 classes מתוך ה- 4 התקבלו ביצועים די טובים ועבור ה- class הרביעי "O" (other) התקבלו ביצועים פחות טובים כפי שהיה באימון עצמו. ניתן לראות שמקרים של AFib זוהו באופן יחסית טוב של 83.1%.

6. **זמן אימון:** תחילה הרצנו את הקוד בהפעלה מרחוק על שרתי האוניברסיטה, דבר שגזל זמן רב מאוד והיה איטי מאוד בעיקר בגלל זמני ההשהיה שנבעו מהעברת הנתונים בהפעלה מרחוק. מכיוון שהאלגוריתם מעט "כבד", ה- data מרובה ביותר והפעלה מרחוק מוגבלת בעקבות זמני התקשורת ורוחב הפס של העברת המידע, ההרצות עבור אימון המערכת לקחו שעות רבות מאוד ואף חצאי יממות, במהלך לילות שלמים ואף יותר.
לאחר מכן עברנו לעבוד על מחשב PC ביתי ללא שליטה מרחוק והדבר קיצר את זמני ההרצה לעשרות דקות או מספר שעות בודדות. ברגע שלא היינו תלויים יותר בקצבי התקשורת מרחוק, הדבר קיצר את זמני ההרצה לעומת שליטה מרחוק פי 3 ואף יותר. המחשב הביתי עליו הרצנו היה בעל CPU בלבד ולא GPU. אילו היה זה מחשב "חזק" יותר ובעל GPU (מעבד גרפי), זמני ההרצה של אימוני הרשת היו יורדים לדקות בודדות או לכל היותר עשרות דקות. נתוני המערכת של המחשב עליו ביצענו את הרצות אימוני הרשת הינם:

מערכת

מעבד:	Intel(R) Core(TM) i5-8400 CPU @ 2.80GHz 2.81 GHz
זיכרון מותקן (RAM):	8.00 GB (7.88 GB ניתנים לשימוש)
סוג מערכת:	מערכת הפעלה של 64 סיביות, מעבד מבוסס x64
עט ומגע:	אין קלט עט או קלט מגע הזמינים עבור צג זה

במערכת הראשונה שאימנו הרצנו 8 אפוקים מתוך 10 בזמן של 822 דקות ו-3 שניות (כמעט 14 שעות) והגענו ל-4273 איטרציות.

מכיוון שהתוצאה בכל מקרה לא היתה טובה מספיק וגם הגענו לפלאטו (בנוסף לזמן ההרצה שגם ככה היה רב מאוד והקצב האיטי להרצת רשת זו) החלטנו לעצור את האימון של הרשת.
לאחר שעברנו למערכת השנייה (אינפוט שונה ורשת זהה) שאימנו הן התוצאות השתפרו והן קצב הריצה היה גבוה יותר וזמני ההרצה התקצרו משמעותית.
האימון השני היה לאורך 30 אפוקים מלאים, לקח זמן של 41 דקות ו-49 שניות, משמעותית הרבה יותר מהיר. האימון כלל 1830 איטרציות. לאימון זה נתנו להגיע עד הסוף ולסיים את 30 האפוקים שתכננו מלכתחילה.
לאחר מכן גם ניסינו להריץ את אותו האימון עם אותם הפרמטרים למספר אפוקים גדול יותר ולא נראה שינוי ניכר באיכות התוצאות (פעם אחת הגרף נהיה "פלאטו" עבור ה-Accuracy ופעם אחת אף ירד – אוברפיטינג). לכן בחרנו להציג את האימון כפי שהרצנו עם 30 אפוקים.

7. **שימוש באוגמנטציה:** בנוסף לדאטא שסופק, עשינו גם שימוש באוגמנטציה. מתוך 8409 סיגנלי ה- ECG שהתקבלו לאחר השוואת אורכי הסיגנלים ל- 9000 דגימות (כפי שפורט בשאלה 4), מתקבל כי 59.2% מהסיגנלים הם נורמליים, כאשר מנגד רק 8.7% הם פרפור עליות ו- 0.4% בלבד מהסיגנלים הם רעש. בעקבות כך, מספיק שהמודל יסווג את כל הסיגנלים כנורמליים או כ- other ונקבל accuracy (נכונות) גבוהה יחסית, אך כמובן שאין זו המטרה וסיגנלים חשובים של AFib לא ייחשבו. לכן, על מנת להימנע מכך, נבצע את האוגמנטציה באופן הבא:

i. נשאף לכך שיתקיים שיוויון בין כמות הסיגנלים פר קטגוריה – התפלגות אחידה, כך שלא תהיה נטייה של הרשת לאף קטגוריה ספציפית.

ii. לאחר חלוקת הדאטא לסטים של אימון וואלדיציה, נבצע השוואת התפלגות בכל סט ע"י שכפול של האותות תחת הקטגוריות: (A), (O) ו- (~) מספר כזה של פעמים, כך שמספר הסיגנלים החדש יהיה שווה למספר הסיגנלים בקטגוריית ה- (N). לדוגמא, אם קיבלנו 4478 סיגנלי נורמל ו- 659 סיגנלי פרפור עליות בסט האימון, יחס של בערך 7:1, נקח את 4473 הסיגנלים הנורמליים הראשונים ולאחר מכן נשכפל 7 פעמים את 639 הסיגנלים הראשונים מפרפור העליות.

iii. תוצאות ההתפלגות לאחר אוגמנטציה הן:

א. עבור סט האימון:

```
summary(YTrain)

A      4473
N      4473
O      4470
~      4470
```

ב. עבור סט הואלידציה (בדיקה):

```
summary(YValid)

A      497
N      498
O      498
~      498
```

אוגמנטציה מסוג זה הינה דרך להרחיב את הדאטא ללא איסוף נתונים חדשים בפועל ועשוי לשפר בצורה משמעותית את ההסתמכות על מדד הנכונות של הרשת.

8. **הצעות לשיפור השיטה:**

i. הוספת אלגוריתם לזיהוי QRS – מעבר להפעלת FFT והפקת תועלת מפיצ'רים בתדר, ניתן להוסיף פרמטר נוסף לפיו תלמד המכונה – RR intervals לדוגמא.

ii. הוספת אלגוריתם לזיהוי גל P – אם ישנו דגש על מציאת AFib, אלגוריתם לזיהוי גל P (שאינו קיים כפי שקיים באות נורמלי) יכול להיות מועיל. אם זאת, עלולה להיווצר רגישות לאותות שסיווגם AFib.

* (i+ii) ניתנים ליישום ע"י אימון Unsupervised CNN שתלמד לזהות את הפיצ'רים, ואז להזינם ל- RNN לביצוע הקלסיפיקציה.

** עבור (iii+iv) – ביצענו ניסיונות רבים של שינויי פרמטרים כמו ה- learning rate, חלוקת המיני באצ'ים או אפילו הגדלת סט האימון וקיבלנו תופעת overfitting.

iii. הקטנת קצב הלמידה בצורה משמעותית או בצורה הדרגתית – עלול לסייע למנוע אוברפיט (להתחיל מקצב גבוה יחסית ולרדת בהדרגה). לשם כך נדרש להריץ על מחשב "חזק" יותר (עם GPU), אחרת האימון יתבצע בצורה איטית ביותר.

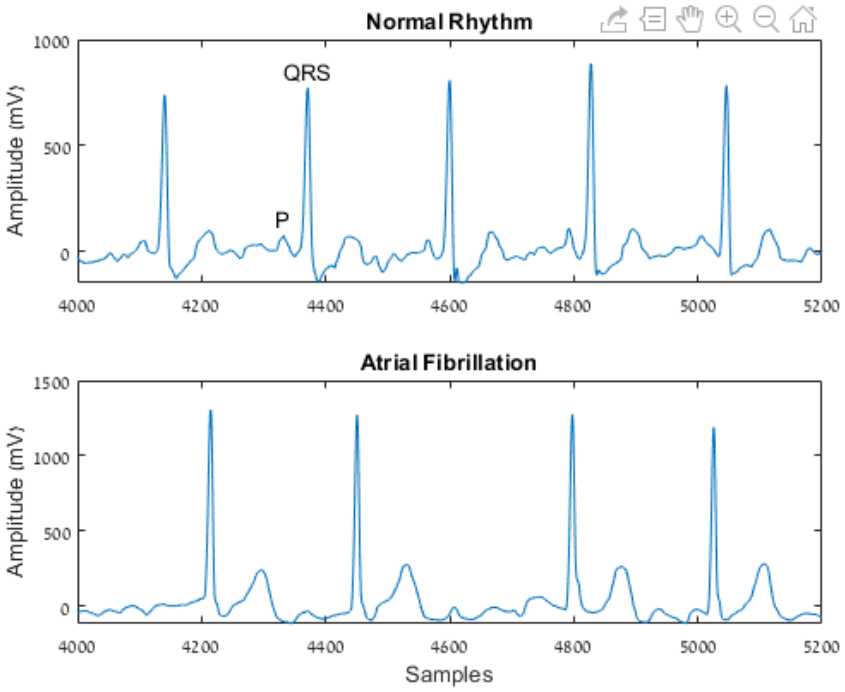
iv. ביצוע Dropout כפי שנלמד בהרצאה (כטכניקת רגולרזיציה).

v. שיפור סיווג סיגנלי (O)[2] – לטעמנו, כדי לחשוב על לפרוט מעט יותר את סיגנלי ה- O, כלומר במקום קטגוריית O אחת, ליצור מספר קטגוריות O, נניח אחת עבור ברדיקרדיה, אחת עבור בעיות בחדרים וכו', שכן נראה כי למערכת שלנו יש בעיה לסווג סיגנלים מסוג זה.

[2] מדובר פחות בשיפור של השיטה, אך ייעול כללי שעלה כמסקנה מהתוצאות.

נספח א':

להלן דוגמא לאות ECG נורמלי ולאות ECG המאפיין פרפור עליות, מתוך הדאטא הנתון:



נספח ב':

הוראות להפעלת הקוד:

דרישות המערכת:

חובה

גרסת Matlab עדכנית (אנחנו עבדנו עם Matlab 2019b), גירסת ה- Matlab חייבת לכלול את ה- ToolBoxes הבאים:

1. Data processing
2. Deep learning

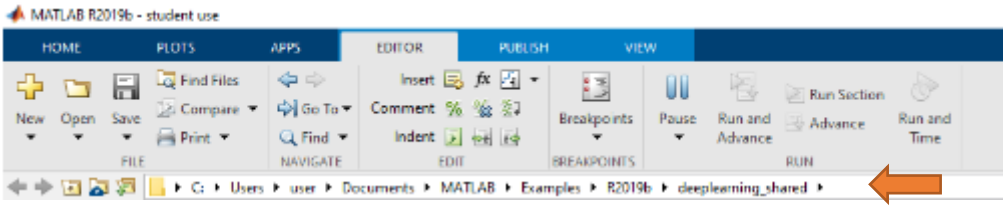
מומלץ

עבור האצת זמן ריצת התכנית, מומלץ להשתמש במחשב עם GPU. אם משתמשים במחשב בעל GPU, כדאי שיהיה גם את ה- ToolBox :

3. Parallel Computing

הפעלת הקוד, ביצוע האימון ובדיקת קבצי ה- Test:

1. יש לבצע unzip לתיקייה שהוגשה.
2. בתוך התיקייה, מופיעה תיקיית Data - יש לשים את 8528 הקבצים איתם מאמנים את הרשת וכמו כן את קובץ ה- REFERENCE.CSV בתוכה. בתוך תיקיית ה- Test, יש לשים את הקבצים (סיגנלים) איתם רוצים לבחון את הרשת ואת קובץ הרפרנס שהפעם יקרא REFERENCE_TEST.CSV. יש לוודא כי שמות קבצי ה- CSV נקראים בדיוק כך. כמו כן, קבצי הסיגנלים של הטסט צריכים להיות מאותו פורמט כמו קבצי הנתונים.
3. יש לפתוח את המאטלב ולוודא כי ה- Path (חץ כתום למטה) מנותב לתיקיית ההגשה (לאחר unzip). לדוגמא: C://users//Ariel//FinalProjectECG_060774908_312178999.



4. כעת ישנן שתי אפשרויות להרצת הקוד:
   a. לפתוח את הקובץ ClassifyECGSignalsLSTM.m.
      i. ניתן להריץ run section על כל section בנפרד, כדי לקבל את הגרפים השונים המופקים לאורך התהליך בזמן אמת, וכן כדי לראות את ה- training progress שמציג את גרף הנכונות והשגיאה בזמן אמת.
      ii. ניתן להריץ run all ולקבל את כל הפלוטים והתוצאות בסוף הריצה.
   b. לפתוח את הקובץ ClassifyECGSignalsLSTMMLX.mlx.

i. האפשרות הראשונה היא הרצה של כל section בנפרד ע"י כפתור ה-run section. יש לבצע זאת section לאחר section כאשר כל אחד מה- sections יבצע קטע רלוונטי בתכנית ויציג את התוצאות הרלוונטיות לאותו ה-section. יש לזכור שלאחר סיום הרצת ה-section יש לסמן את ה-section הבא ורק לאחר מכן ללחוץ run section.
ii. הרצת כל התכנית ע"י לחיצת Run all. במצב זה כל התכנית תרוץ מתחילתה ועד סופה.

5. במהלך ההרצה יוצג תהליך האימון והוולידציה על גרף (בהנחת run section, אחרת יוצג בסוף).
6. לאחר האימון והואלידציה, התכנית תעבור מיד להרצת הקבצים עבור ה- Test אשר בסיומה תופיעה תוצאת ה- Accuracy הרלוונטית ולאחריה ה- Confusion Matrix הרלוונטית.