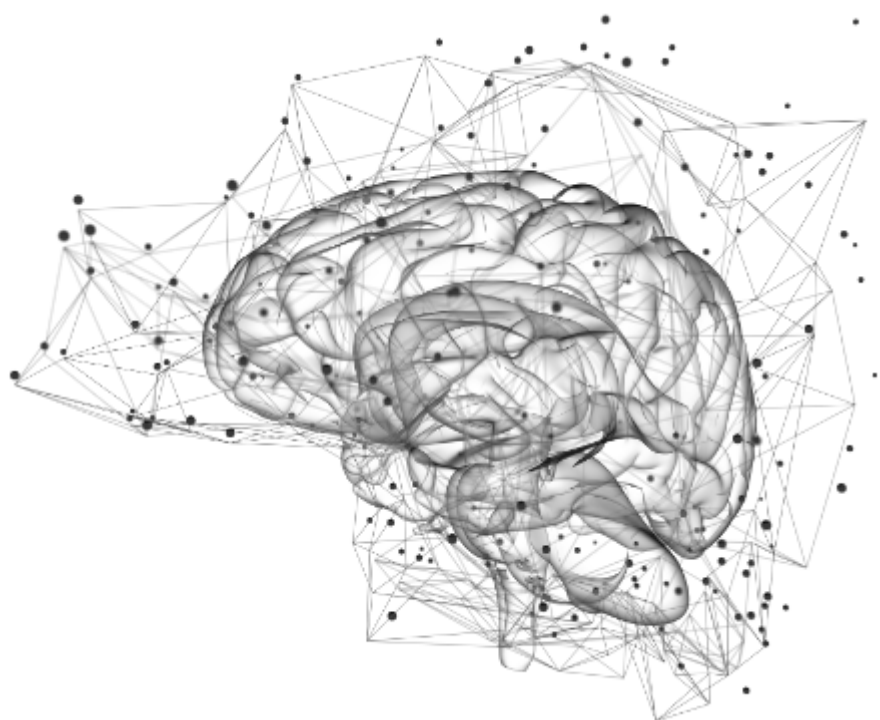


Sep 8, 2024

83-695

## פרויקט סיום רשתות נוירונים



מוגש ע"י:

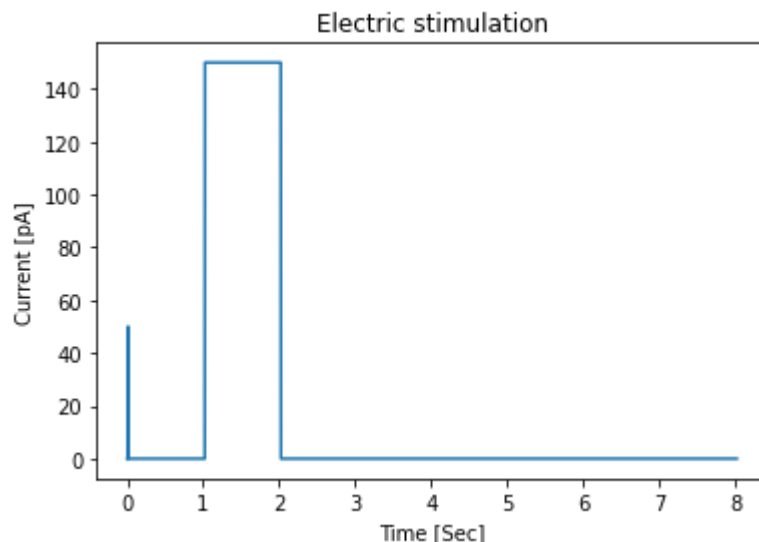
אור שחר - 206582017  
יובל בן שלמה - 312117666

## חלק א

### 1. קראו על נוירונים aspiny ו-aspiny בקצרה מהם סוגי נוירונים אלו

- נוירונים aspiny ו-aspiny הם שני סוגים עיקריים של נוירונים במוח, ההבדל העיקרי ביניהם זה מבנה הדנדריטים שלהם ובהתאם לכך בתפקידים שהם ממלאים במוח.
- נוירונים מסוג Spiny- נוירונים בעלי מבנה של דנדריטים עם סיגמואידים (spines), כאשר הסיגמואידים הם שלוחות קטנות הנמצאות על הדנדריטים ומשמשות כמקומות שבהם מתבצעת העברת מידע בין נוירונים.
- נוירונים אלה נפוצים במיוחד באזורים של המוח כמו קליפת המוח וגרעיני הבסיס (basal ganglia). הסיגמואידים על הדנדריטים שלהם מאפשרים מספר רב של קלטים סינפטיים, כך שניתן לעבד מידע רב.
- הנוירונים מעורבים בעיקר בתהליכים של עיבוד מידע מורכב, למידה וזיכרון, תנועות רצוניות ועיבוד תחושת.
- נוירונים מסוג Aspiny הם בעלי מבנה של דנדריטים ללא סיגמואידים (spines). זה אומר שהדנדריטים שלהם הם חלקים יותר וכוללים פחות אתרי מגע סינפטיים.
- הם נפוצים יותר במערכת העצבים המרכזית והם מעורבים בעיקר בתפקידים מעכבים במוח, כמו ויסות פעילות נוירונלית והגנה על המוח מפני פעילות יתר, כמו בקרת פעילות נוירונים אחרים כדי למנוע פעילות יתר שעשויה לגרום לפרכוסים.

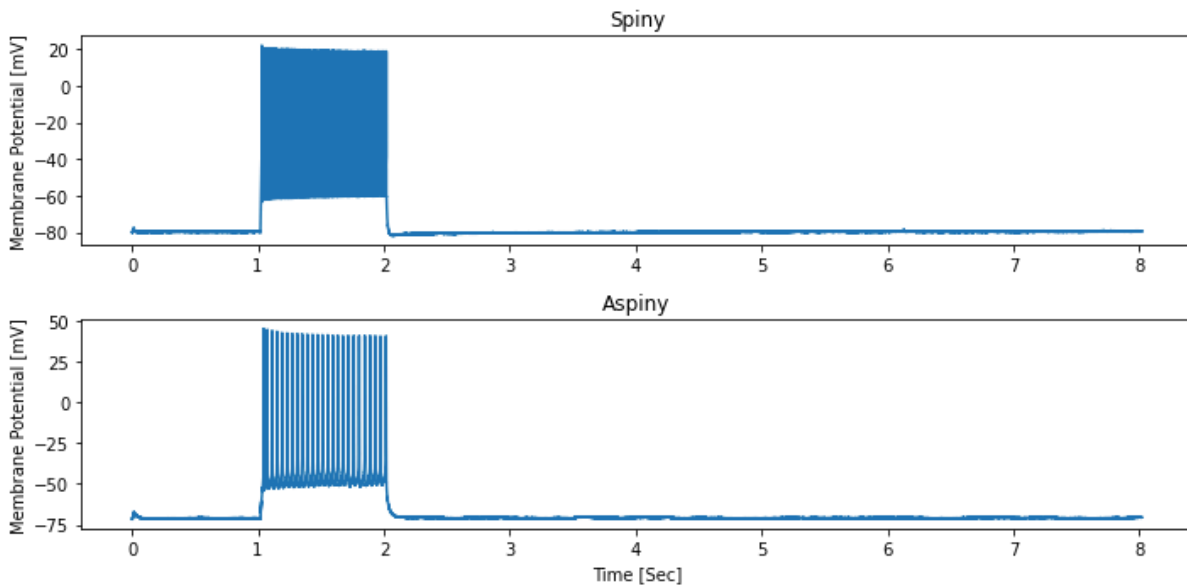
### 2. הציגו את אות הגירוי החשמלי כתלות בזמן. תארו את הגירוי החשמלי בקצרה



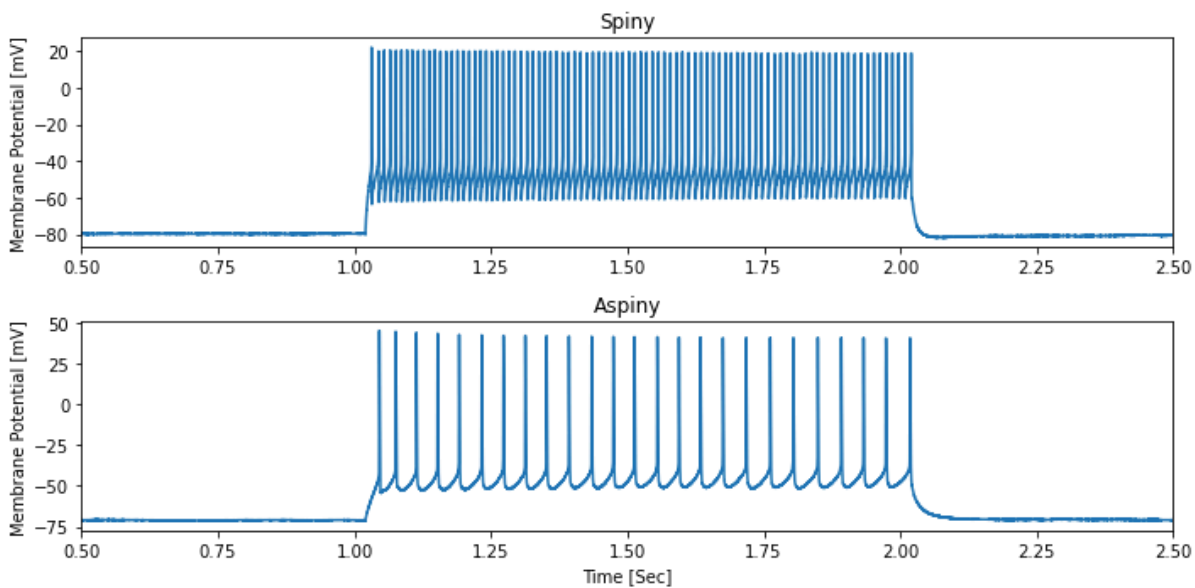
\*הקפיצה ב 0 היא Bug.

- ניתן לראות שהגירוי מתחיל בערכים נמוכים (אפסיים) ועולה בחדות לאחר שנייה לערך מקסימלי של 140pA, לאחר שניה נוספת ערך הגירוי יורד בחזרה בבת אחת לערך אפסי ונשאר כך לאורך המשך הניסוי, אורך הניסוי הכולל הינו 8sec.
- כלומר הנוירונים קיבלו גירוי ממושך ולא גירוי רגעי.

3. הציגו את הפעילות החשמלית של הניורון ה-spiny והניורון ה-aspyne בתגובה לגירוי החשמלי. תארו בקצרה את התגובה:

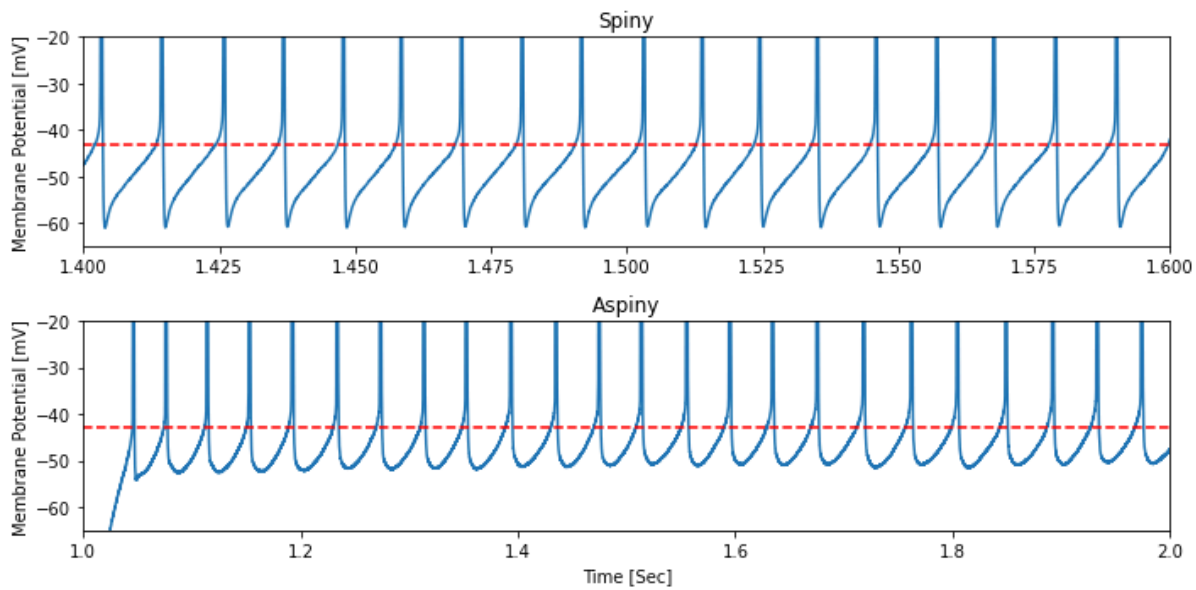


נעשה זום אין לטווח זמן שבין 0.5-2.5 שניות כדי שנוכל לראות טוב יותר את הספייקים:



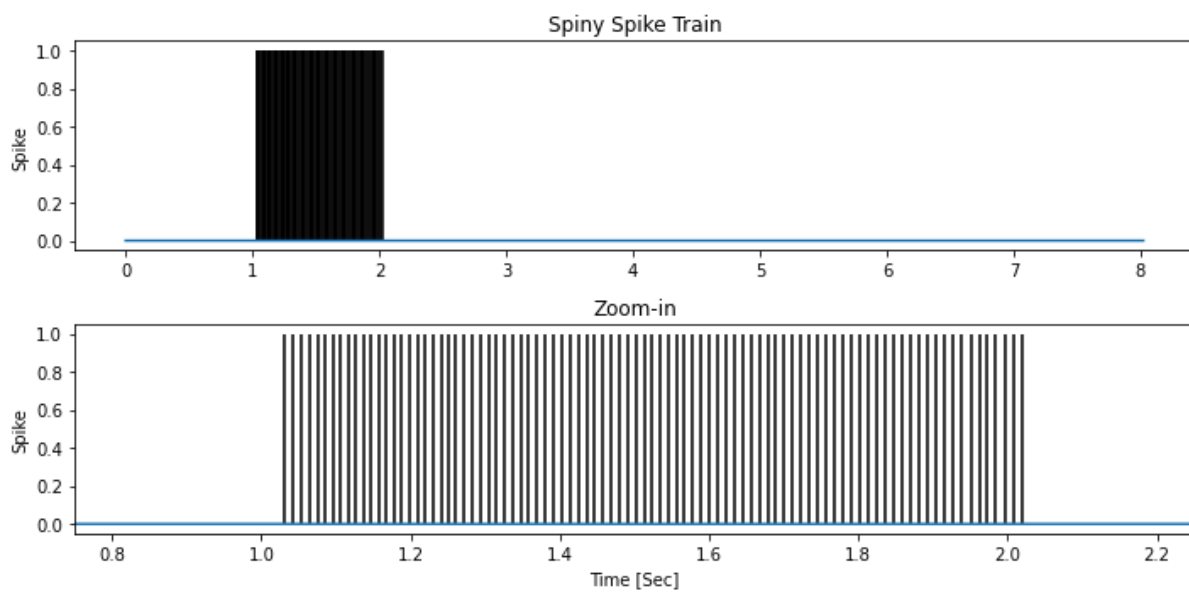
ניתן לראות כי הגירוי החשמלי גרם לפוטנציאל פעולה ב2 סוגי הניורונים. בניורון ה-spiny ניתן לראות כי פוטנציאלי הפעולה קרו בתדירות גבוהה יותר והספייקים צפופים יותר לעומת הניורון ה-aspyne שבו קצב הירי איטי יותר והספייקים מרווחים יותר. בנוסף, אפשר לראות כי פוטנציאל המנוחה ב2 סוגי הניורונים שונה, ב-spiny הוא בסביבות ה-79mV וב-aspyne בסביבות ה-72mV. כמו כן, הפוטנציאל החשמלי המקסימלי אליו מגיעים הניורונים שונה אף הוא, ב-spiny המקסימום הוא 18mV וב-aspyne המקסימום הוא 45mV.

4. חשבו והציגו train spike של שני הנוירונים בתגובה לגירוי החשמלי. מהו ערך הסף שמעליו מתרחש פוטנציאל פעולה בנוירונים אלו?

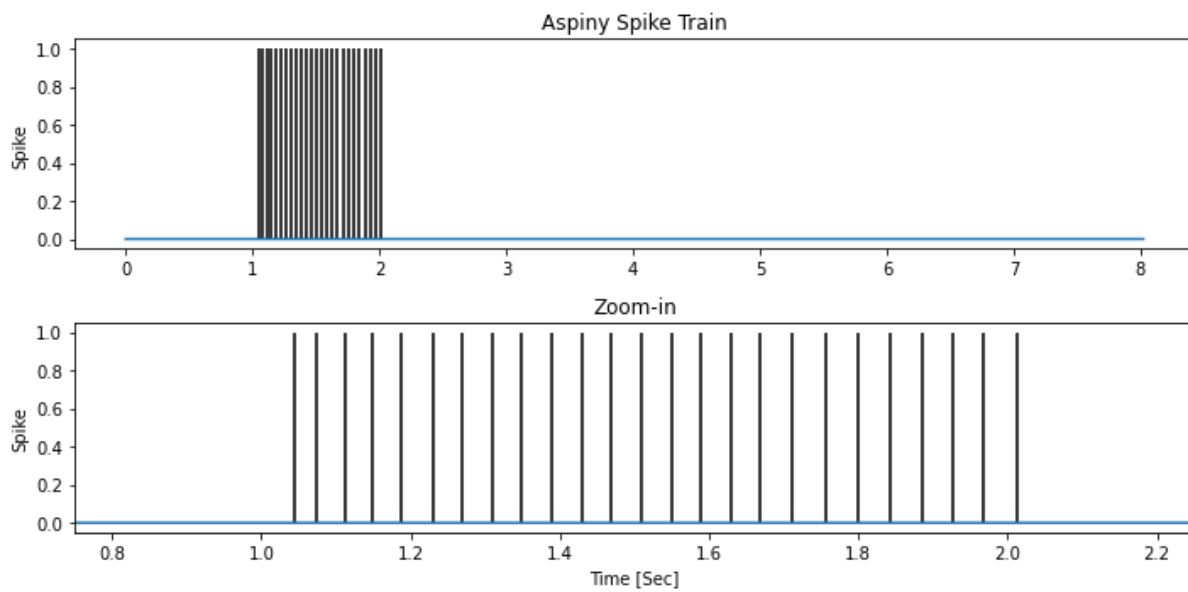


- קבענו את ערך הסף לפי העין להיות: -45mV.
- הוספנו התחשבות refractory period, כדי להימנע מתוצאות שווא.

עבור הנוירון הspiny:



### עבור הspiny:



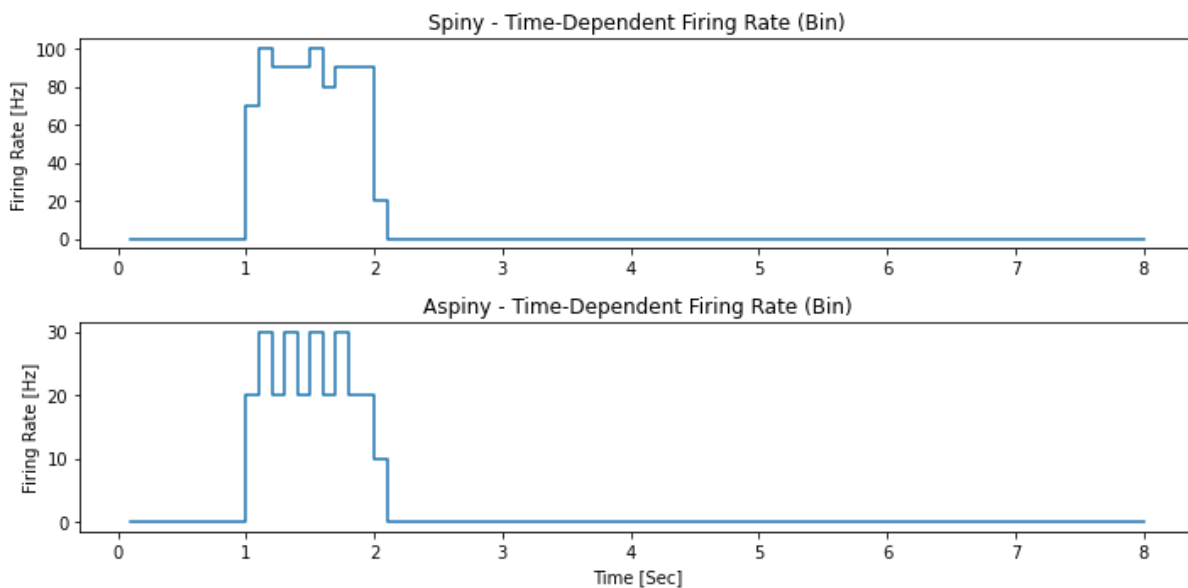
- הקווים האנכים מתארים את נקודות הזמן בהן התרחשו פוטנציאלי פעולה, ניתן לראות שהתוצאות מתאימות לסעיף הקודם, עבור נירון spiny תדירות פוטנציאלי הפעולה גבוהה יותר, ועבור נירון Aspiny ישנם פחות פוטנציאלי פעולה ולכן צפיפות הקווים האנכיים קטנה יותר.

5. חשבו והציגו בגרף את ה- **rate firing dependent time** עבור כל אחד מסוגי הניורונים. השתמשו ב- **bins** וב- **kernel/window** שנראים לכם מתאימים כדי שפונקציית קצב הירי תתאר בצורה טובה את תגובת הניורונים לגירוי החשמלי. הסבירו את בחירתכם.

- חישוב על ידי **bins**: נחלק את הזמן למינימום  $\text{bin}=100\text{ms}$ , כלומר נבחן קבוצות זמן של  $100\text{ms}$  ונסכום כמה **spikes** מופיעים בכל חלון זמן. לאחר מכן נבצע חלוקה של כמות הספייקים לחלק לאורך ה- **bin** כלומר  $\text{bin\_spike\_count}/\text{delta\_bin}$  ובכך נקבל את היחס של ספייקים לשניה. נבצע תצוגת מדרגות הממחישה את כמות הספייקים לשניה שחושבה פר **bin**. נשתמש בנוסחה

$$r(t) = \frac{1}{\Delta t} \int_t^{t+\Delta t} d\tau \langle \rho(\tau) \rangle$$

- התוצאות עבור bins:

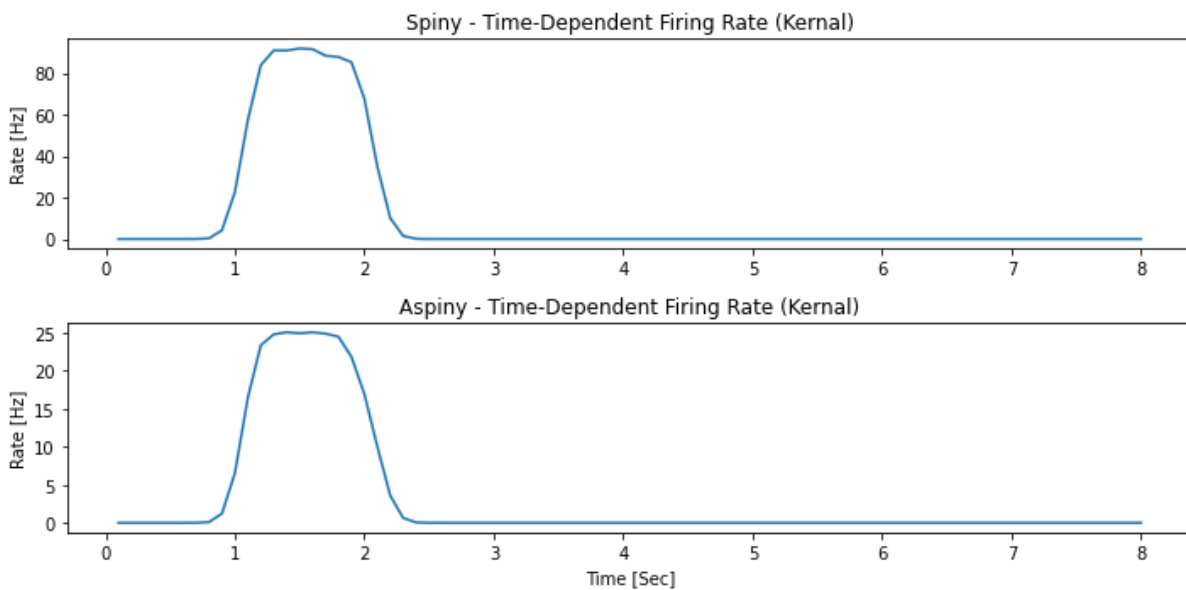


- החלקת Gaussian kernel:

$$w(\tau) = \frac{1}{\sqrt{2\pi}\sigma_w} \exp\left(-\frac{\tau^2}{2\sigma_w^2}\right)$$

גאוסיאן קרנל נותן יותר משקל לערכים שסביב נקודה  $t$  ופחות משקל לערכים הרחוקים מנקודה  $t$ . הסיגמה מייצגת את רוחב ההחלקה של ה- **Gaussian Kernel**, ומתורגמת ממונחי זמן למונחי דגימות בהתאם לתדר הדגימה. ככל שסיגמה קטנה יותר, רוחב החלון עליו נסתכל הוא צר יותר, מה שמשפיע על "רמת ההתחשבות" בערכים מסביב לנקודה. אנחנו בחרנו  $\sigma = 1$ ,

• התוצאות עבור kernal:



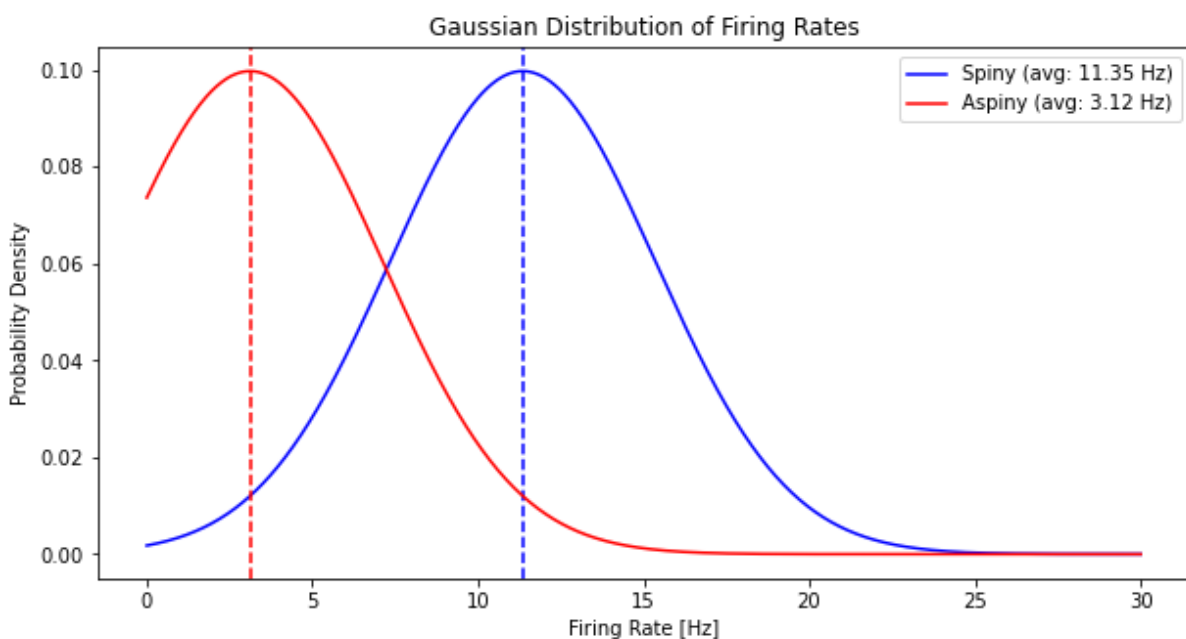
6. מהו קצב הירי הממוצע של כל אחד מהניורונים בתגובה לגירוי החשמלי? הניחו שקצב ירי זה מאפיין את התגובה של כל אחד מסוגי הניורונים. סביב ממוצע זה, צרו התפלגות גאוסיאנית של ההסתברות לקצבי ירי בתגובה לגירוי עבור כל אחד מסוגי הניורונים. השתמשו בסטיית תקן לבחירתכם עבור ההתפלגויות כך שיש ביניהן חפיפה חלקית. מצאנו שקצב הירי ההמוצע עבור כל אחד מהניורונים הוא:

Average Firing Rate - Spiny: 11.35 [Hz]

Average Firing Rate - Aspiny: 3.12 [Hz]

א) הציגו את ההתפלגויות בגרף:

סטית התקן שבחרנו כך שתהיה חפיפה,  $\sigma_r = 4$ :



(ב) מהו ה-  $d'$  discriminability עבור ההתפלגויות שיצרתם? הראו כיצד חישבתם.

- נשתמש בנוסחה הבאה שלמדנו בהרצאה:

$$d' = \frac{\langle r \rangle_+ - \langle r \rangle_-}{\sigma_r}$$

כאשר בחרנו את סטיית התקן להיות 4Hz.

$\langle r \rangle_+$  מייצג את ממוצע קצב הירי עבור נירון spiny

$\langle r \rangle_-$  מייצג את ממוצע קצב הירי עבור נירון Aspiny

נציב ונקבל:

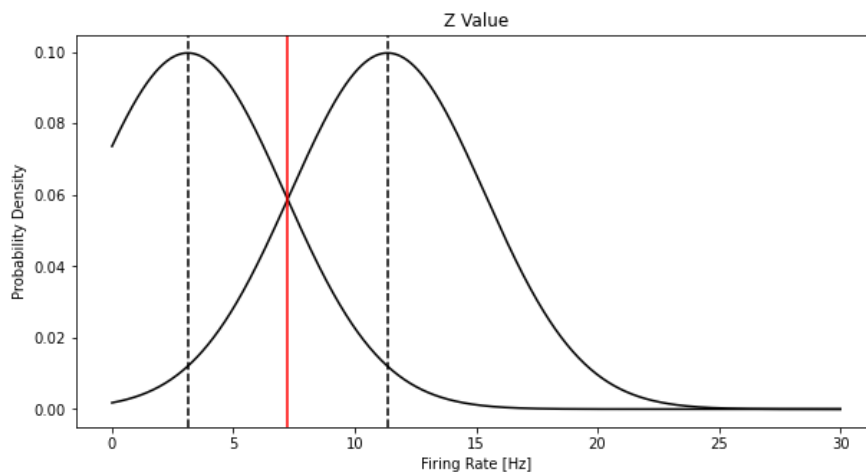
$$d' = \frac{11.35 - 3.12}{4} = 2.0575$$

7. בחרו ערך סף  $z$  כלשהו המשמש לסיווג בין שני סוגי הנוירונים. נתחו את הביצועים של המסווג: הציגו matrix confusion וחשבו מדדים רלוונטיים כגון accuracy, recall וכו'.

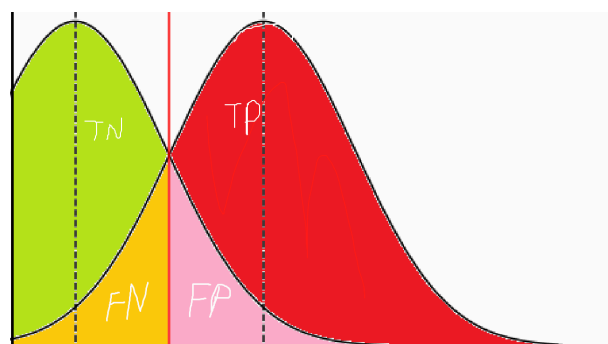
- נבחר את ערך סף  $z$  ע"י חישוב ממוצע בין שני הממוצעים, כלומר:

$$z = \frac{11.35 + 3.12}{2} = 7.23 [Hz]$$

מיוצג ע"י הקו האדום בגרף הבא:



כאשר החלוקה מתבצעת בצורה הבאה:





● מטריצת הבלבול שקיבלנו:

	Predicted Spiny (Positive 1)	Predicted Aspiny (Negative 0)
Actual Spiny (Positive 1)	TP: 0.85012235	FN: 0.14987765
Actual Aspiny (Negative 0)	FP: 0.19411281	TN: 0.80588719

● המטריצה מציגה ארבעה ערכים עיקריים:

- (a) **True Positive**: מספר הניבויים הנכונים שמזהים כנכונים (נוירון מסווג כ-Spiny והוא אכן Spiny).
- (b) **True Negatives**: מספר הניבויים הנכונים שמזהים כשליליים (נוירון מסווג כ-Aspiny והוא אכן Aspiny).
- (c) **False Positives**: מספר הניבויים השגויים שמזהים כנכונים (נוירון מסווג כ-Spiny אך הוא Aspiny).
- (d) **False Negatives**: מספר הניבויים השגויים שמזהים כשליליים (נוירון מסווג כ-Aspiny אך הוא Spiny).

הפרמטרים הסטטיסטיים:

- **Accuracy**: מדד שמחשיב את אחוז הניבויים הנכונים (גם חיוביים וגם שליליים).

$$Accuracy = \frac{TP+TN}{TP+TP+FN+FP} = 0.83$$

- **Recall**: מדד שמחשיב את אחוז הניבויים החיוביים הנכונים מתוך כלל התוויות החיוביות האמיתיות.

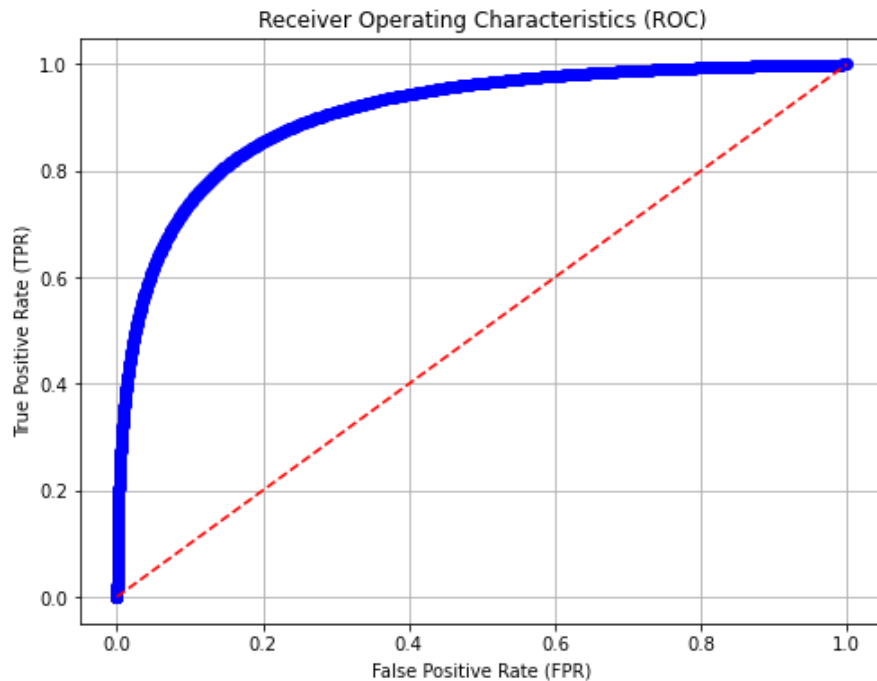
$$Recall = \frac{TP}{TP+FN} = 0.85$$

- **Precision**: מדד שמחשיב את אחוז הניבויים החיוביים הנכונים מתוך כלל הניבויים החיוביים.

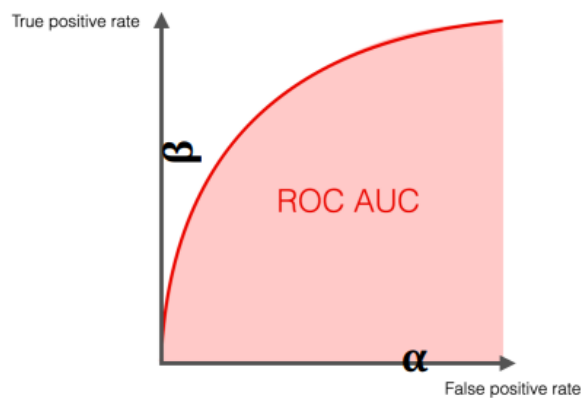
$$Precision = \frac{TP}{TP+FP} = 0.81$$

8. חשבו והציגו ROC (Receiver operating characteristics) עבור הסיווג בין שני סוגי הנזירים על בסיס המסווג הפשוט. מהו ה- Area under the curve שהתקבל ומה משמעותו? איזה שינוי בנתונים שעל בסיסם חיבתם אותו עשוי להביא ל-AUC גדול או קטן יותר?

הגרף שלנו:



- AUC הוא השטח מתחת לעקומה ROC, הוא מדד שמשקף את איכות הביצועים של מודל הסיווג הבינארי שלך על כל טווח ערכי הסף האפשריים. הערך שלו נע בין 0-1, כאשר  $AUC=1$  אומר שהמודל מסווג בצורה מושלמת את הנתונים בלי טעויות,  $AUC=0.5$  הוא מודל אקראי (כמו הטלת מטבע) וכאשר  $AUC>0.5$  המודל נחשב למסווג לא טוב.



## נחשב את הAUC:

ROC AUC: 0.9092061513544283

- שינויים בנתונים שיכולים להשפיע על הAUC:
  - מודל מאומן יותר יפיק AUC טוב יותר.
  - אם יש רעש בנתונים או שהתוויות לא מדויקות זה יכול לפגוע בAUC.
  - ככל ששתי הקטגוריות נבדלות זו מזו בצורה ברורה יותר יהיה למודל קל יותר להבחין ביניהן, וה-AUC יעלה.
  - הגדרת הסף ה-ROC Curve נבנה עבור ערכי סף שונים, שמחלקים את תחום ההסתברויות של המודל כדי לקבוע האם דגימה מסווגת כחיובית או שלילית. בחירת סף אופטימלי יכולה לשפר את הביצועים של המודל עבור תרחישים ספציפיים, אבל ה-AUC מייצג את הביצועים של המודל לאורך כל תחום הסף האפשרי.
  - כאשר יש חוסר איזון בין מספר הדגימות החיוביות לשליליות (למשל, יותר נזירים מסוג אחד לעומת השני), זה עלול להשפיע על ה-AUC. במקרים כאלה, המודל עלול להעדיף את הקטגוריה הדומיננטית ולייצר תחזיות שגויות לדגימות מהקטגוריה המיעוטית.
- ניתן לשפר את הAUC על ידי ניקוי נתונים, טיפול בנתונים לא מאוזנים, טיפול בערכים חסרים.

---

## חלק ב

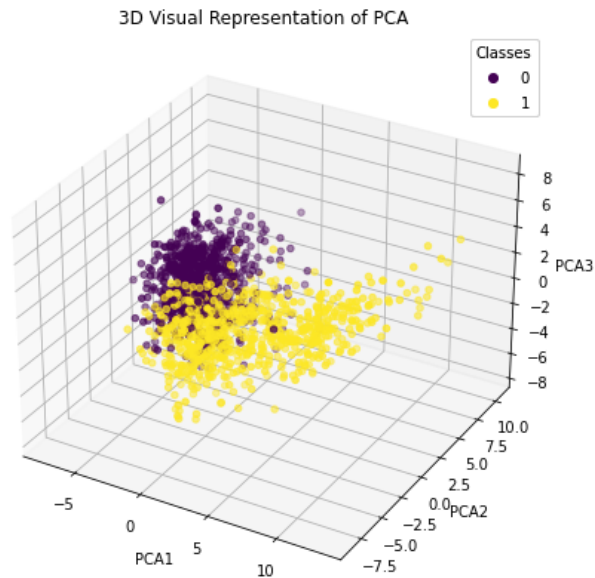
### שאלה 1:

בחלק זה נשתמש בשיטות למידה בלתי מפוקחת על מנת לעשות אנליזה לנתונים. ממשו והריצו PCA- Analysis Component Principal עבור הורדת מימד של הנתונים לשלושה מימדים.

- PCA הוא אלגוריתם שנועד להקטין מימדים ולמצוא תכונות מרכזיות בנתונים, תוך שמירה על כמה שיותר מהמידע המקורי. הרעיון הוא להמיר את הנתונים לרכיבים חדשים, בלתי תלויים, שנקראים רכיבים עיקריים, שכל רכיב כזה מכיל כמה שיותר מהשונות (Variance) בנתונים המקוריים.
- נרצה להשתמש ב-PCA מכמה סיבות: כאשר הנתונים מורכבים מהרבה משתנים ותכונות, הורדת המימדים מאפשרת להציג את הנתונים בצורה פשוטה יותר וקלה יותר להבנה אך כן נשמור על כמה שיותר מידע. בנוסף שימוש ב-PCA עוזר לזהות דפוסים ומגמות בנתונים.
- שלבי ה-PCA
  1. חישוב ממוצע של כל עמודה בנתונים (כלומר לכל משתנה שלנו), החסרת הממוצע מכל ערך בעמודה (זה מייצר מרכז הנתונים סביב ה-0).
  2. מציאת מטריצת הקווריאנס שמייצגת את הקשר בין המשתנים.
  3. חישוב הערכים העצמיים שמייצגים את גודל השונות של כל רכיב עיקרי, חישוב הווקטורים העצמיים, שמייצגים את כיוון הרכיבים העיקריים.
  4. בחירת הרכיבים העיקריים לפי הערכים העצמיים שלהם, ככל שהערך העצמי של רכיב גבוה יותר הוא מסביר יותר מהשונות בנתונים.
  5. השלכת הנתונים על המרחב החדש שנוצר ע"י הרכיבים שבחרנו.

א. הציגו את שלושת הצירים של ה-PCA באמצעות `plot scatter` והציגו בצבעים שונים את כל אחד מסוגי הנוירונים.

הגרף שלנו: כאשר '0' מייצג נוירון מסוג `spiny`, ו'1' מייצג נוירון מסוג `Aspiny`

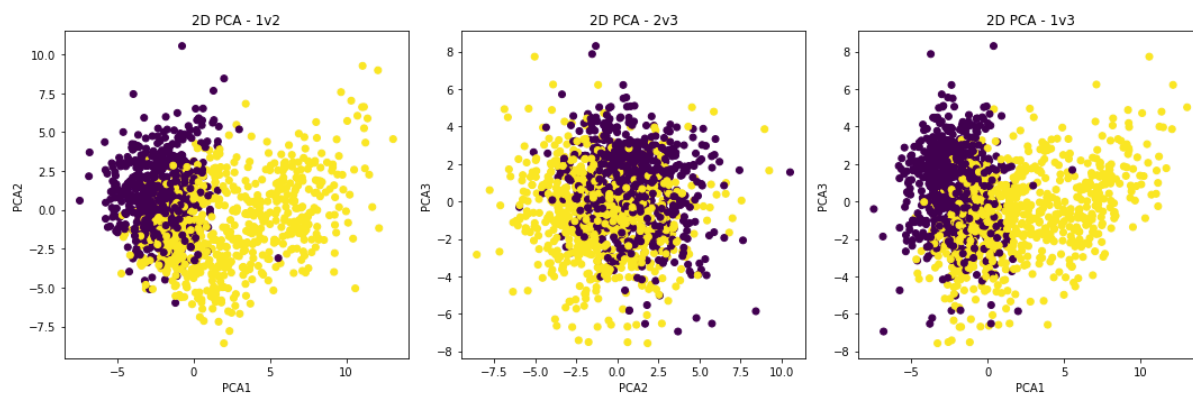


## ב. מה מייצגים הצירים של PCA?

הצירים מייצגים את הרכיבים העיקריים (Principal Components) המבטאים את הכיוונים שבהם יש את הכי הרבה שונות בנתונים המקוריים. ה-PCA לוקח את כל המשתנים בנתונים-כלומר התכונות של הנירונים, ומצמצם אותם למספר קטן יותר של צירים חדשים, כך שכל ציר (רכיב עיקרי) מסביר כמה שיותר מהשונות בנתונים.

- PCA1 (הציר הראשון): מייצג את הכיוון שבו יש את השונות הגדולה ביותר בנתונים. הכיוון שבו הנתונים המקוריים נפרדים בצורה הטובה ביותר. זה הרכיב העיקרי שמסביר את מרבית השונות במערכת הנתונים. לדוגמה, אם יש תכונה אחת מאוד דומיננטית, היא תופיע בצורה חזקה בציר הזה.
- PCA2 (הציר השני): מייצג את הכיוון שבו יש את השונות השנייה בגודלה, זה הציר שמאונך לציר הראשון. ציר זה מסביר את מה שנותר מהשונות בנתונים אחרי שהציר הראשון לקח את החלק הגדול ביותר. לכן, PCA2 מסביר שונות נוספת שקיימת בנתונים, אבל פחותה מזו שהוסברה על ידי PCA1.
- PCA3 (הציר השלישי): מייצג את הכיוון שבו יש את השונות השלישית בגודלה, והוא מאונך לשני הצירים האחרים. כמו PCA2, גם ציר זה מסביר שונות שנותרה אחרי שהרכיבים הקודמים כבר הסבירו את רוב השונות בנתונים.
- השונות בנתונים מתארת עד כמה הערכים של כל תכונה משתנים או נעים סביב ממוצע כלשהו. כשאנו מבצעים PCA, המטרה היא למצוא את הכיוונים (הצירים) שמסבירים כמה שיותר מהשונות הזו. הצירים החדשים (PCA1, PCA2, PCA3) הם כיוונים במרחב הנתונים שמהווים צירוף לינארי של התכונות המקוריות, אבל הם נבחרים כך שיכילו כמה שיותר מהשונות בנתונים.

## ג. האם יש צירים המפרידים טוב יותר מאחרים בין סוגי הנירונים?



בגרפים שמוצגים בתמונה, ניתן לראות את ההתפלגות של שני סוגי הנוירונים (המוצגים בצבעים שונים) בשלושה צירופים של רכיבי PCA. ניתן להסיק ש- PCA1 (הרכיב הראשון) הוא הציר שמפריד בצורה הברורה ביותר בין שני סוגי הנוירונים. יש הבחנה יחסית טובה בין הקבוצות כאשר משווים את PCA1 ל-PCA2 או PCA3.

PCA2 ו-PCA3 פחות מוצלחים בהפרדה בין הקבוצות, אפשר לראו שקיים ערבוב של הצבעים, כלומר, הם פחות מסבירים את השונות שקשורה להבדל בין הנוירונים.

כלומר- PCA1 הוא הרכיב שמסביר את השונות הגדולה ביותר בנתונים ומפריד בצורה הטובה ביותר בין סוגי הנוירונים.

**ד. בכמה צירים של PCA צריך להשתמש כך שניתן יהיה לשחזר את המידע המקורי עד שגיאה של 1% מבחינת ה Mean squared error ?**

MSE הוא מדד סטטיסטי שמשמש להערכת איכות המודל או איכות השחזור של הנתונים. הוא נותן אינדיקציה עד כמה הנתונים המשוחזרים קרובים לערכים המקוריים. הוא מחושב בצורה הבאה:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

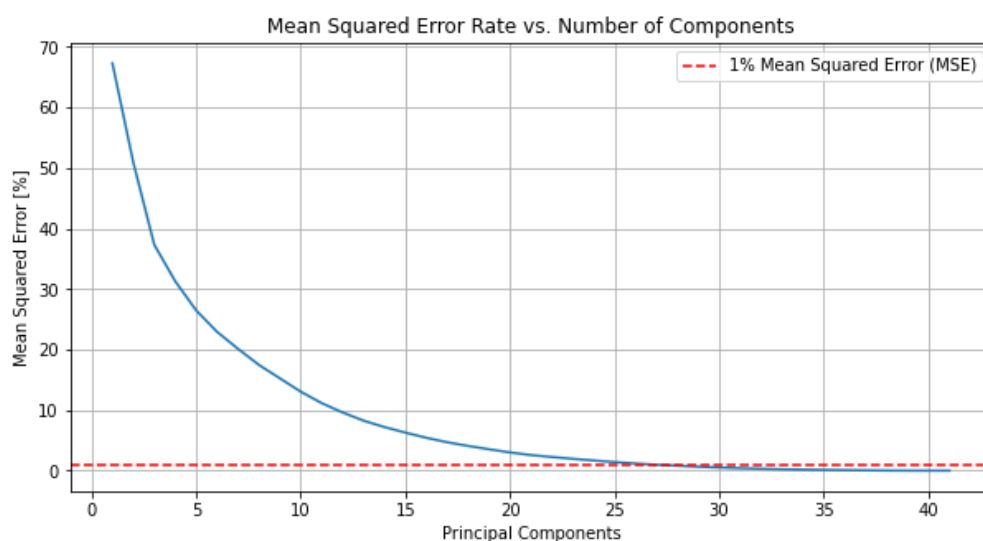
כאשר n זה מספר הדגימות,  $y_i$  זה הערך האמיתי של הדגימות ו  $\hat{y}_i$  מייצג את הערך המשוחזר של דגימה  $y_i$ .

ככל שה MSE נמוך מייצג שחזורים מדויקים יותר.

איך זה מתקשר ל PCA? כאשר אנו רוצים לשחזר את הנתונים המקוריים באמצעות מספר מצומצם של רכיבים עיקריים, נשתמש ב-MSE כדי להעריך עד כמה השחזור מצליח לשמור על המידע המקורי. כל עוד מוסיפים רכיבים עיקריים (Principal Components), השחזור משתפר, וה-MSE פוחת, כי הרכיבים הנוספים מסבירים יותר מהשונות בנתונים.

התוצאות שלנו:

Number of components needed for less than 1% error: **27**



## שאלה 2:

בנו רשת autoencoder עם שלוש שכבות חבויות ופונקציית אקטיבציה מסוג:  $\text{relu}$  אם מספר הזהות האחרון של בני/ות הזוג הוא זוגי ו  $\tanh$  אחרת. בחרו את מימד ה  $\text{bottleneck}$  להיות 3.

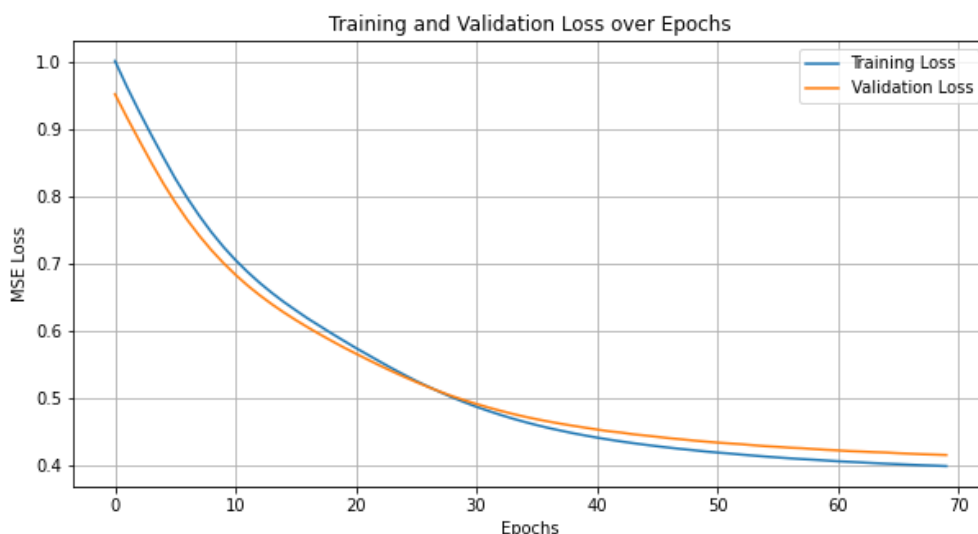
- רשת autoencoder היא סוג נוסף של רשת נוירונית מלאכתותית שמטרתה כמו PCA היא ללמוד ייצוג דחוס של נתונים ולאחר מכן לשחזר אותם בצורה שתהיה קרובה יותר לנתונים המקוריים.  
הרשת בנויה מ  $\text{encoder}$  שתפקידו לצמצם את המידע ולהמיר את הנתונים למרחב מימדים קטן יותר, ו  $\text{decoder}$  שתפקידו לקחת את הייצוג הקטן ולשחזר ממנו את הנתונים המקוריים.
- הרשת בנויה משכבת הקלט, שכבות חבויות -בשלב זה הרשת מבצעת צמצום הדרגתי של מימדי הנתונים, שכבת  $\text{bottleneck}$ -השכבה שמכילה את הייצוג הדחוס של הנתונים,  $\text{decoder}$ -שם המידע עובר תהליך הפוך ומוחזר לגודל המקורי שלו, במטרה לשחזר את הנתונים המקוריים.
- מספרי התז שלנו הם  $6+7=13$  אי זוגי, לכן נבחר בפונקציית אקטיבציה מסוג  $\tanh$ .  
פונקציית אקטיבציה מסוג טנגנס היפרבולי נראית כך:

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1}.$$

הפונקציה ממפה את הערכים של  $x$  לטווח בין -1:1. היא מאפשרת לרשת ללמוד ערכים חיוביים ושליילים. הפונקציה מספקת גרדיאנטים עבור כל ערכי הקלט מה שמאפשר לרשת ללמוד טוב יותר. בנוסף הפלט שלה יהיה ממורכז סביב האפס וזה עוזר בתהליך הלמידה להפיץ את הנתונים בצורה טובה יותר בתוך הרשת עצמה.

1. אמנו את הרשת על סט האימון, הציגו את גרף השגיאה (עבור לוס של MSE) לאורך האימון.

\*נריץ את החישוב ל70 אפוקים.





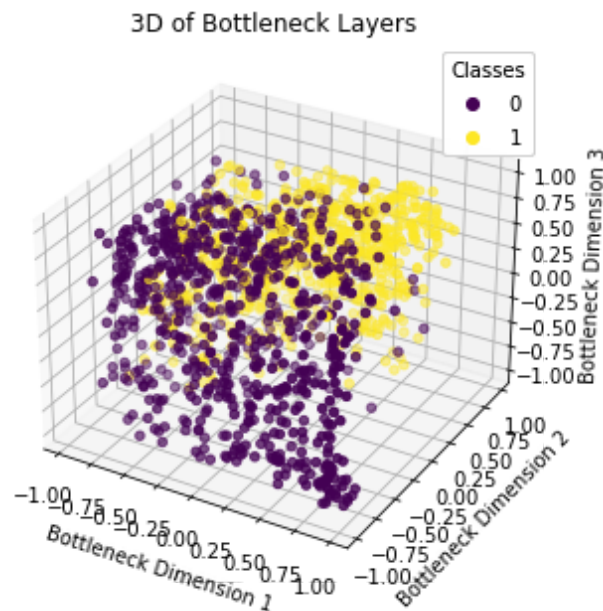
- ניתן לראות בגרף שלנו את ה-MSE לשני סוגי הסטים, הכחול עבור סט האימון והכתום עבור סט האימות.

כמצופה, ב-epochs הראשונים ה-MSE גבוהה, כלומר השחזור נתונים יהיה לא קרוכ למידע האמיתי, אך עם התקדמות ה-epochs ה-MSE קטן עבור שני הסטים, כלומר המודל לומד לשפר את ביצועיו בכל אפוק, כלומר השחזור שלו משתפר.

קווי ה-MSE עבור סט האימון וסט האימות קרובים מאוד זה לזה, במיוחד לקראת סוף האימון. זה סימן טוב מאוד, שכן זה אומר שהמודל לא סובל מבעיה של overfitting. בשלב כלשהו סביב 50 אפוקים ה-MSE מתייצב יחסית ואינו יורד עוד משמעותית, כלומר המודל מגיע למיצוי מסוים של הלמידה, בשלב זה ניתן להפסיק את האימון.

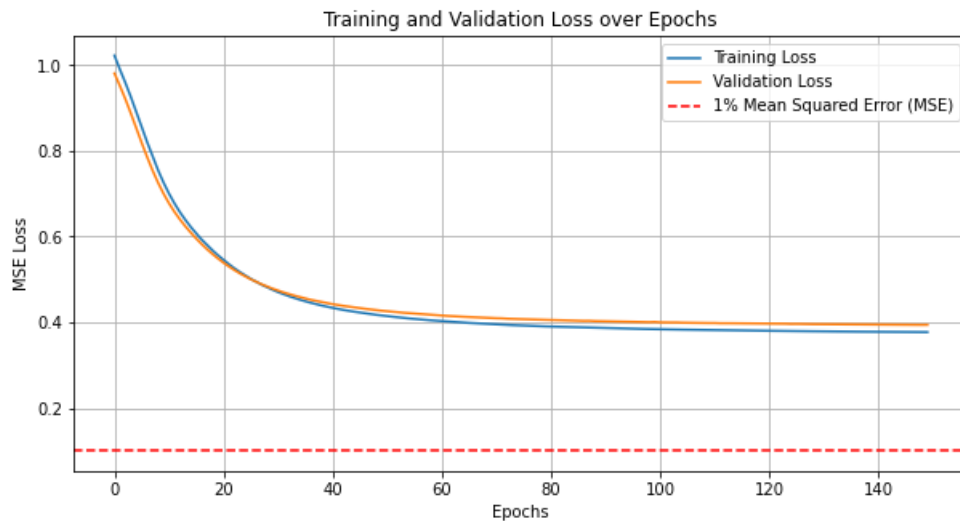
## 2. הציגו את הדאטה בייצוג הנמוך שנלמד ע"י רשת ה-autoencoder. כלומר הציגו plot scatter של שלושת הצירים ב-bottleneck.

הגרף שלנו: כאשר '0' מייצג נירון מסוג spiny, ו'1' מייצג נירון מסוג Aspiny



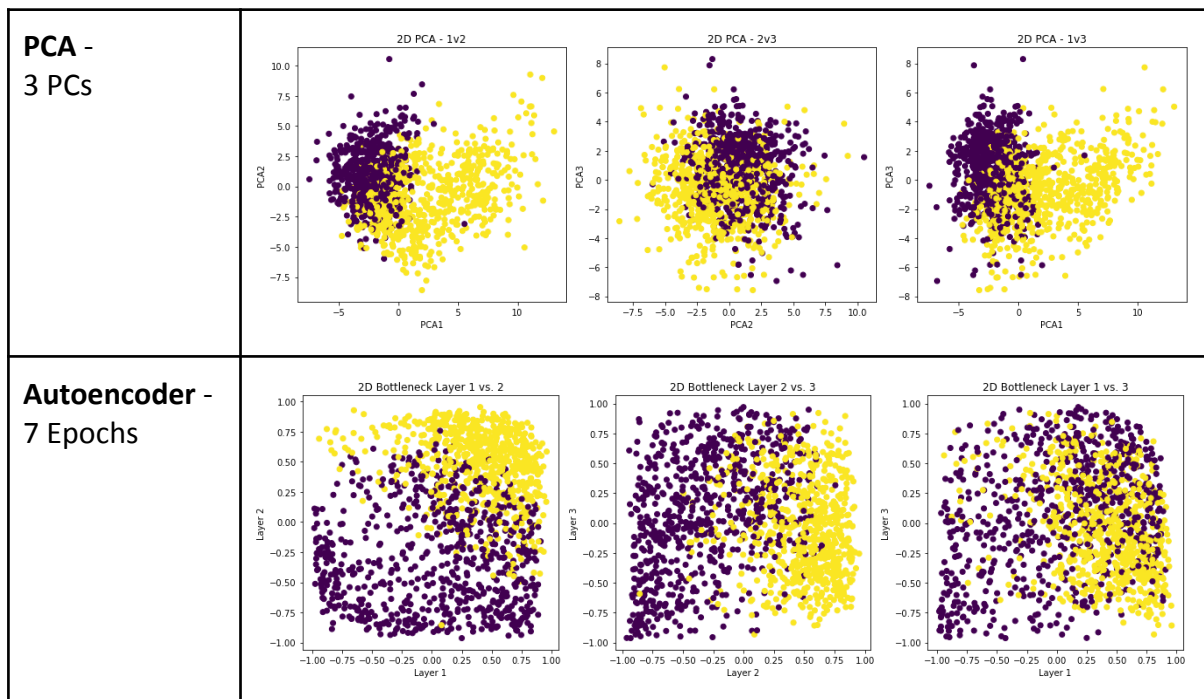
## 3. האם רשת ה-autoencoder למדה ייצוג טוב יותר להפרדת סוגי התאים לעומת PCA? נמקו או הדגו.

נבצע השוואה בין השיאות, עבור ה-Autoencoder נבצע כ-150 אפוקים:



ראינו בחלק הקודם שגרף MSE עבור PCA מצליח לחצות את 1%, אך ניתן לראות פה הגרף עבור האוטו-אנצודר מתייצב סביב  $MSE=4\%$  כבר לאחר 50 אפוקים. כלומר, רשת PCA מבצעת ייצוג טוב יותר להפרדת סוגי התאים.

גם בשיטת השוואה חזותית בתצורת דו-מימד, ניתן לראות כי הסיווג של האוטו-אנצודר פחות מוצלח אפילו בהשוואה ל-PCA עם 3 רכיבים עיקריים:



## חלק ג

בחלק זה תאמנו מסווגים כדי להפריד בין נירון spiny ונירון aspiny. השתמשו בקובץ הנתונים מהחלק השני והשתמשו בעמודה המגדירה את סוג הנירון כדי להגדיר את ה label. כלומר אל תשתמשו בעמודה זו כמאפיין לאימון המסווגים. חלקו את הדאטה ל test-train באופן אקראי ככה שסט האימון מהווה 80% מהדוגמאות.

1. אמנו מודל רגרסיה לוגיסטית על סט האימון (train) ובחנו אותו על סט הבחינה (test). רגרסיה לוגיסטית היא שיטה לניתוח והסקת מסקנות כאשר משתנה המטרה הוא בינארי. זה מודל סיווג שמשמש בפונקציה סיגמואיד כדי להעריך את ההסתברות שמקרה מסוים שייך לאחת משתי הקטגוריות. (א) נתחו את השגיאות של המודל כלומר הציגו matrix Confusion וחשבו recall Precision כאשר הנירון ה spiny יוגדרו כמחלקה ה"חיובית".  
התוצאות שלנו:

Logistic Regression	Predicted Spiny (Positive 1)	Predicted Aspiny (Negative 0)
Actual Spiny (Positive 1)	148	1
Actual Aspiny (Negative 0)	1	135
Precision:	0.9926470588235294	
Recall:	0.9926470588235294	

2. אמנו perceptron על סט האימון ובחנו על סט הבחינה. ה-Perceptron הוא יחידת חישוב שמנסה לסווג נתונים לשתי קטגוריות (סיווג בינארי). הוא מקבל קלט, מבצע שילוב לינארי של הקלטים, ומחליט אם הקלט שייך למחלקה אחת (0) או למחלקה שנייה (1) על בסיס פונקציית סף.  
התוצאות שלנו:

Perceptron	Predicted Spiny (Positive 1)	Predicted Aspiny (Negative 0)
Actual Spiny (Positive 1)	148	1
Actual Aspiny (Negative 0)	8	126
Precision:	0.9922480620155039	
Recall:	0.9411764705882353	

### א) השוו בין שני המודלים מי הכליל טוב יותר?

- מבחינת תוצאות ה-Precision: בשני המודלים התוצאה זהה כמעט לחלוטין (0.992).
- ההבדל המשמעותי בין המודלים הוא ב-Recall
- Perceptron זיהה פחות דוגמאות חיוביות (Recall של 0.94), כלומר, הוא פספס 8 דוגמאות של spiny.
- הרגרסיה לוגיסטית הייתה מדויקת יותר בזיהוי הדוגמאות החיוביות, עם Recall של 0.99, ופספסה רק דוגמה אחת.
- המסקנה היא שהרגרסיה הלוגיסטית מצליחה יותר ב-Recall, כלומר, היא סיווגה נכונה יותר דוגמאות חיוביות מ-Perceptron והמודל שלה הכליל טוב יותר.

### ב) האם המודל סובל מ-overfitting?

Training Precision:	0.9725776965265083
Training Recall:	0.9432624113475178

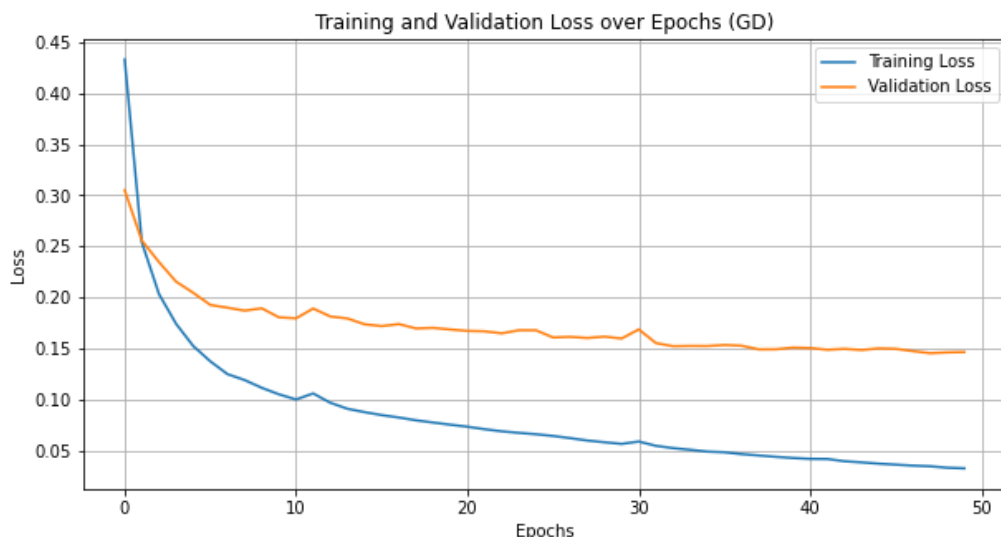
- Training Precision – כ-97% מהתחזיות של המודל שסווגו כ-spiny בסט האימון היו נכונות.
- Training Recall – כ-94% מהדוגמאות שהיו spiny בסט האימון סווגו נכון על ידי המודל.
- כלומר- התוצאות על סט האימון קרובות מאוד לתוצאות על סט הבדיקה. זה מצביע על כך שאין פער גדול בין הביצועים על סט האימון לבין הביצועים על סט הבדיקה, לכן ניתן להניח שהמודל לא סובל מ-Overfitting.

### 3. בנו רשת MLP-preceptron layer Multi על מנת לסווג את שני סוגי הנורונים.

- MLP הוא סוג של רשת נוירונים מלאכותית הבנוי ממספר שכבות: שכבת קלט, שכבות חבויות ושכבת פלט. מטרתו היא ללמוד לייצג קשרים בין הקלט לפלט על ידי העברת המידע דרך השכבות והפעלת אקטיבציות לא ליניאריות.

4. אמנו את הרשת עם GD- Gradient Descent על סט האימון, הציגו את גרף ה loss עבור האימון ועבור סט ולידציה בגודל 10% מסט האימון.

- Gradient Descent הוא האלגוריתם המשמש לאימון רשתות נוירונים. המטרה שלו היא למצוא את המשקלים האופטימליים עבור הנוירונים כך שהמודל יצליח לחזות נכון את הפלט מהקלט. כדי לעדכן את המשקלים, ה-Gradient Descent משתמש בחישוב הנגזרת של פונקציית ההפסד ביחס למשקלים.

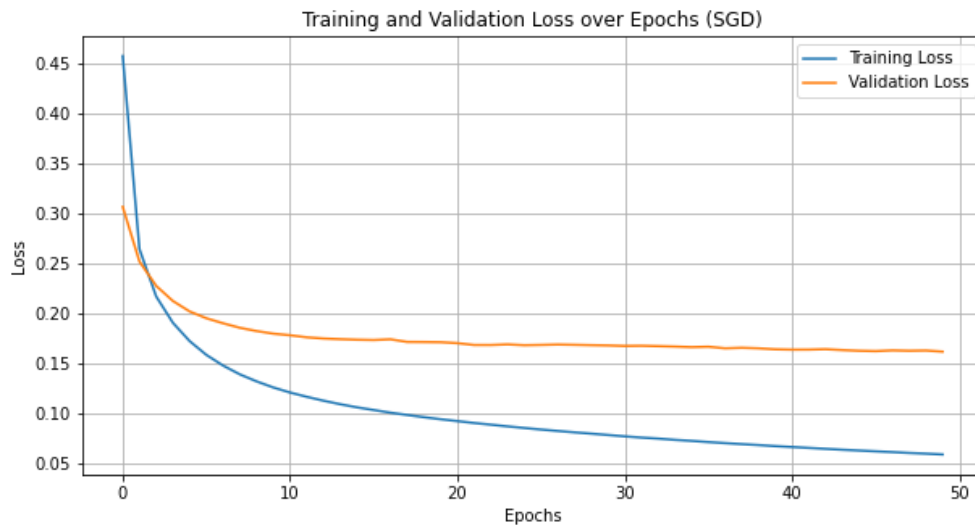


(א) נתחו את השגיאות של המודל כלומר הציגו matrix Confusion וחשבו recall Precision כאשר הנוירון ה spiny יוגדרו כמחלקה ה"חיובית".  
התוצאות שלנו:

GD MLP	Predicted Spiny (Positive 1)	Predicted Aspiny (Negative 0)
Actual Spiny (Positive 1)	148	1
Actual Aspiny (Negative 0)	1	135
Precision:	0.9926470588235294	
Recall:	0.9926470588235294	

(ב) חזרו על הסעיף הקודם אך עם SGD-Descent Gradient Stochastic עם גודל בטש של 10.

- SGD הוא גרסה משופרת של Gradient Descent שבה האלגוריתם לא מחכה לעבור על כל הדוגמאות בסט האימון לפני שהוא מעדכן את המשקלים. במקום זאת, הוא מעדכן את המשקלים אחרי כל דוגמה בודדת או אחרי כל mini-batch וכך יוצר עדכונים מהירים יותר בתהליך האימון.



### התוצאות שלנו:

SGD MLP	Predicted Spiny (Positive 1)	Predicted Aspiny (Negative 0)
Actual Spiny (Positive 1)	148	1
Actual Aspiny (Negative 0)	1	135
Precision:	0.9926470588235294	
Recall:	0.9926470588235294	

### ג) האם המודל סובל מ overfitting?

Training Precision:	0.982174688057041
Training Recall:	0.9769503546099291

- ניתן לראות ששני הערכים שלנו גבוהים, בנוסף ניתן לראות בגרף שההבדל בין ה validation set וה training set הוא קטן (בערך 0.1) ולכן ניתן להסיק שאין overfitting

ד)מה ההבדל בין המודלים שאומנו בשני הסעיפים הקודמים? איזה מודל מכליל טוב יותר?

ההבדלים בין המודלים:

Stochastic Gradient Descent - SGD	Gradient Descent - GD	
מבוצע אחרי כל דוגמה (או מיני-באטצ')	מבוצע אחרי חישוב על כל הדוגמאות	<b>עדכון משקלים</b>
מהיר יותר לעדכונים כי מבצע עדכונים אחרי כל דוגמה או קבוצה קטנה	איטי יותר לעדכונים כי מחכה עד מעבר על כל הנתונים	<b>מהירות חישוב</b>
פחות זיכרון כי עובד על דוגמה אחת או על מיני-באטצ'	צריך לזכור את כל הדוגמאות לפני העדכון	<b>שימוש בזיכרון</b>
עדכונים רועשים יותר שמובילים לתנודתיות	עדכון מדויק יותר ומייצב את הלמידה	<b>דיוק</b>
מתאים לסטים גדולים מאוד	פחות יעיל כשיש הרבה נתונים	<b>סקלביליות לנתונים גדולים</b>
יותר טוב בהיחלצות ממינימום מקומי	לעיתים נתקע במינימום מקומי	<b>יכולת לצאת ממינימום מקומי</b>

- מכיוון שזמן הלמידה טוב (לא ארוך מידי שגורם לoverfitting ולא קצר מידי שגורם לאי דיוקים) וכמות הנתונים לא גדולה, אז 2 המודלים מסווגים בהצלחה זהה וקשה להחליט מי עדיף במקרה הספציפי שלנו.