

APML EX3

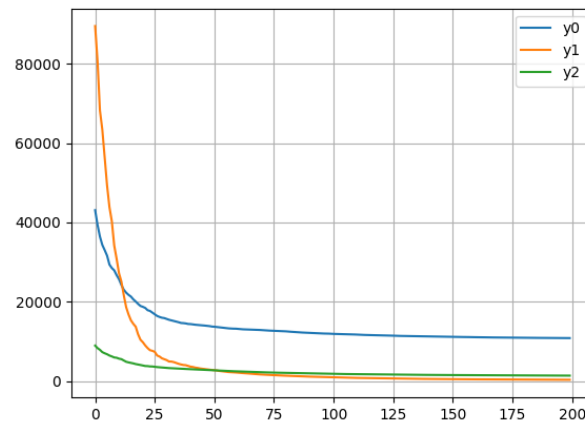
שאלה 2.4

תוצאות המודל הלינארי:

ה loss של הטסט



תוצאות ה convnet :



תחילה ניתן לראות שהמודל הלינארי לא הצליח להתכנס לתוצאה יפה עבור הפונק' y_2 , y_1 . בנוסף אפשר לראות כי עבור הפונקציות הנ"ל גרף השגיאה נותר יציב, רצינו לקרב את המודל הלינארי לפונקציות אבל הפונקציות 1,2 אינן לינאריות לכן הצליח המודל להתקרב אליהן ממש מעט (ניתן לראות ירידה קטנה בהתחלה) אך כיוון שהמודל לינארי נשארת תמיד שגיאה של קירוב בין גרף הפונקציה שהוא מתאר לגרף של הפונקציות 1,2 (אם רוצים לחשוב על הבעיה בצורה גאומטרית, קירוב של גרף קעור בעזרת קווים ישרים).

לגבי y_0 הפונקציה עצמה לינארית ולכן למודל היה קל ללמוד אותה עד כדי מזעור השגיאה כמעט לחלוטין.

כלומר המודל כן מדויק ועובד טוב, השאלה היא איזה פונ' רוצים ללמוד בעזרתו.

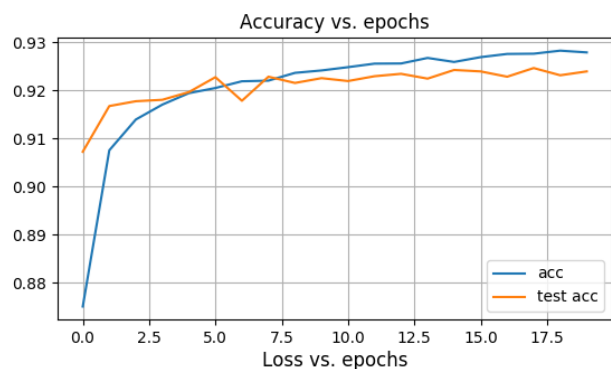
לעומת זאת, convnet יכולה להגדיר משפחה גדולה יותר של פונקציות בעזרת השכבות, בפרט פונקציות שאינן לינאריות כיוון שיש לנו גם פונ' max וגם פונ' relu שאינן לינאריות. ואכן ניתן לראות שמשפחת הפונ' לא לינארית על ידי התבוננות על הגרף של y_0 שהיא פונ' לינארית. הרשת הגיעה לנק' בה היא לא יכולה לצמצם יותר את השגיאה שוב אפשר לחשוב על זה גאומטרית כקירוב של גרף קעור לייצוג של גרף ישר.

בנוסף ניתן לראות כי הפונ' 1,2 הגיעו לשגיאה נמוכה, כפונ' שאינן לינאריות.

שאלה 2.5

המודל הלינארי:

תוצאה סופית : [0.2688840436562896, 0.9239999979734421]



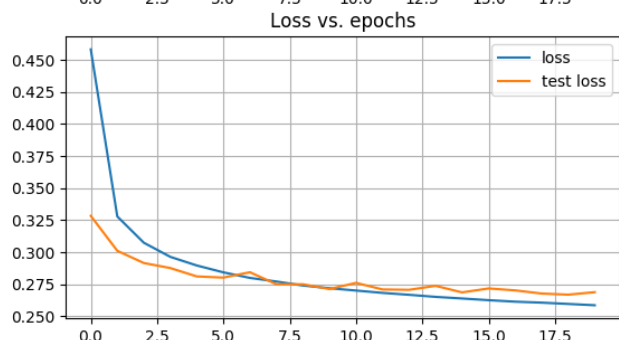
ניתן לראות שהגענו לתוצאה יחסית טובה

עבור שכבת FC אחת.

אפשר לראות שהמודל מצליח להתקרב

לסט האימון אך אינו יציב בטסט לחלוטין

מה שמראה שהלמידה לא מספיק מדויקת.



מודל MLP :

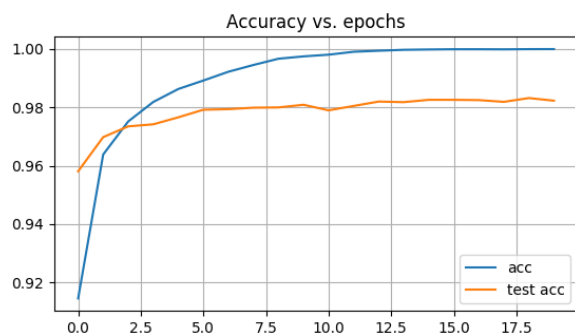
תחילה הוספתי למודל הקודם שכבה נוספת בגודל 10 עם relu וקיבלתי את התוצאה :

score : [0.22259172247955575, 0.9364999988675118]

בבדיקה על העמקת המודל ראיתי שהתוצאה הסופית כמעט ולא משתנה מעומק של יותר מ4 שכבות. לבסוף בחרתי מודל עם 3 שכבות FC בנוסף לאקטיבצית relu. כל שכבה קטנה יותר מקודמתה, בעיקר משיקול של זמני ריצה (כמות משקולות קטנה יותר).

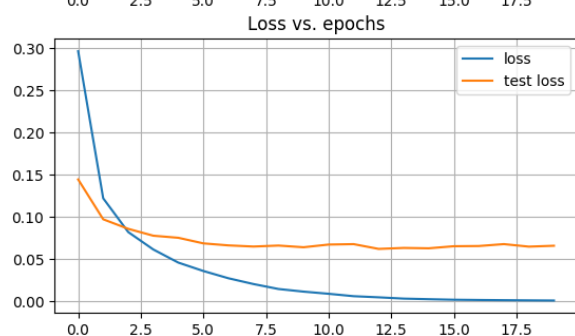
בחישוב loss בחרתי ב - categorical_crossentropy שהיא פונקציה המתאימה למודל שלי שמוציא וקטור הסתברויות אל מול הטסט שהוא וקטור אפסים עם 1 במקום התיג הנכון. בקצרה, הפונ' הנ"ל סוכמת את לוג ההסתברויות כפול הערכים באינדקסים המקבילים בטסט כלומר במקרה שלנו פשוט לוג של הערך באינדקס שמספרו מופיע בתמונה. כלומר אם בהסתברות 1 זה אכן התיג הנכון \leq לוג 1 יוצא ≤ 0 loss=0

הגרף שהתקבל:



ביחד עם התוצאה :

score : [0.06610373745325603, 0.9823000019788742]



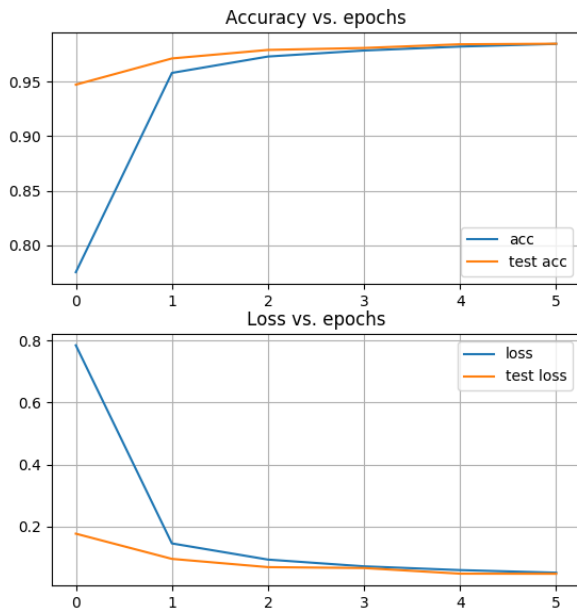
מודל ה Convnet :

גם במודל הזה בחרתי loss של categorical_crossentropy, הפלט של המודל והlabels של האימון נשארו זהים כמו במודלים האחרים לכן אין סיבה לשנות את חישוב הloss שמתאים מאותם שיקולים שהסברתי קודם.

את המודל עצמו בניתי משתי שכבות קונב' בניהן שתי שכבות max-pooling ולבסוף FC לפני ה output הסופי.

משחק עם השכבות מבחינת גדלים ועומק הקרנלים לא משנה כמעט את התוצאה למעט הכבדה על זמני הריצה במודלים עמוקים יותר.

התוצאות :



score: [0.04767881222032884, 0.9847000008821487]

לסיכום:

ניתן לראות שהוספת שכבה אחת + relu הוסיפו מורכבות ואי לינאריות שהעלתה לנו את הדיוק מ 92 ל 98 ופה הייתה בעיקר הקפיצה העיקרית.

במודל הקונב' הוספנו אפילו עוד מורכבות והגדלנו את משפחת הפונקציות שהרשת מסוגלת לתאר, אימנתי על 6 epochs עם batchSize=500

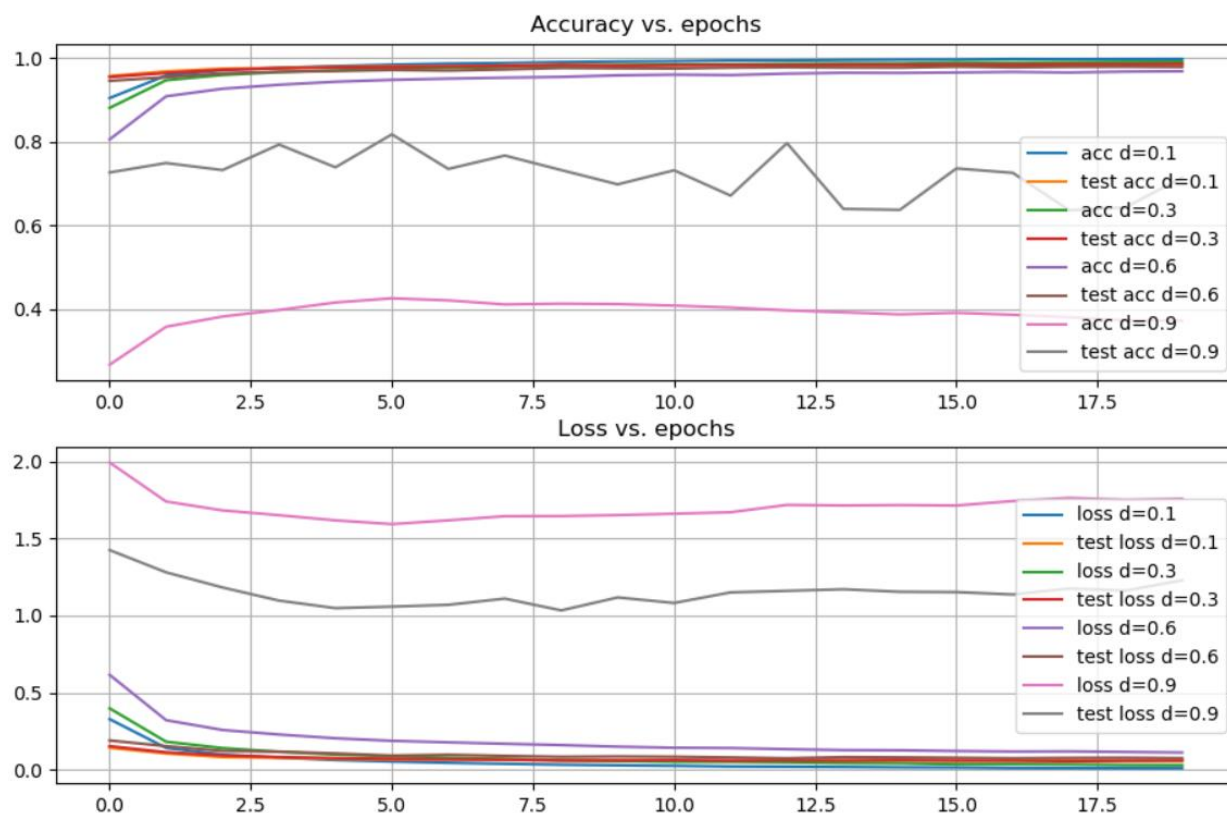
והגעתי לדיוק טיפה יותר גבוהה ברשת הקונב' אפשר גם לראות שיש מעט overfit במודל MLP שלא נראה במודל הקונב' מה שנותן עוד מדד לכך שהלמידה של המודל הזה טובה יותר. בהסתמך על הכמות הקטנה יחסית של איטרציות הלמידה אפשר להסיק כי מודל הקונב' טוב יותר. מה שמתיישב עם מה שלמדנו כי רשתות אלה עובדות טוב יותר על דאטא של תמונות.

חקירת היפר פרמטר :

בחרתי בפרמטר dropout עבור מודל MLP בתקווה לראות השפעה על overfit של הגרף שקיבלתי ושינויים כללים בגרף הלמידה. הוספתי שתי שכבות dropout בין שכבות FC ונתתי להם פרמטר זהה.

בדקתי הורדה של 10, 30, 60 ו90 אחוז מהמשקולות במודל.

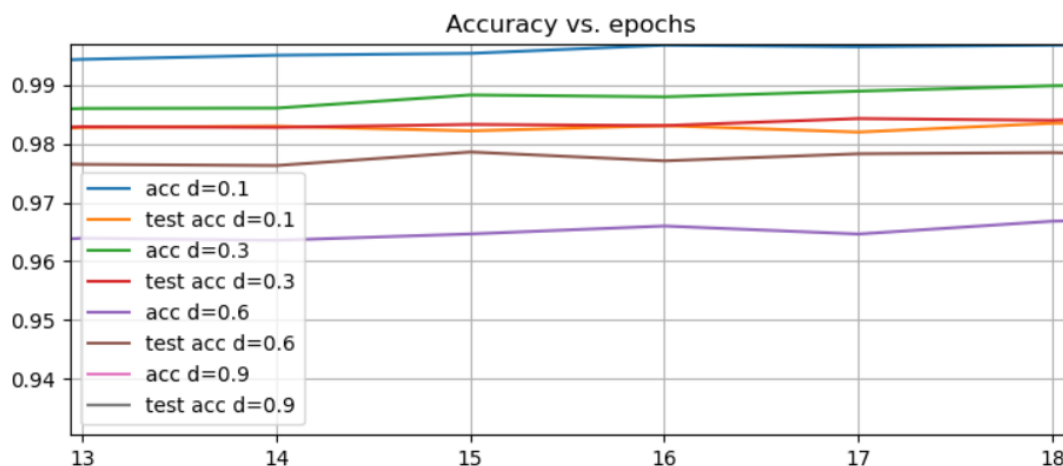
הגרפים שקיבלתי:

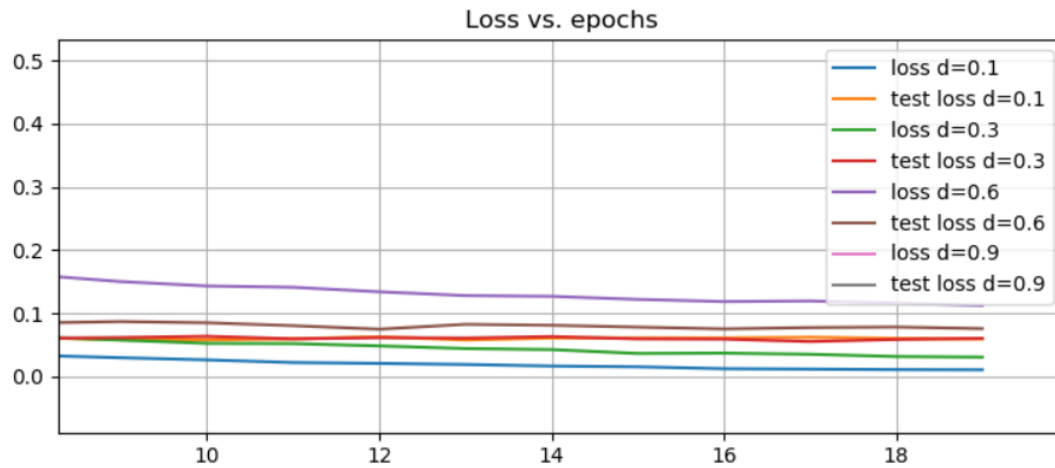


0.1 : score : [0.059138250551443436, 0.9839000016450882]
 0.3 : score : [0.0601337310825096, 0.9851000037789345]
 0.6 : score : [0.0755119291957817, 0.9779000017046928]
 0.9 : score : [1.2289287286996842, 0.7137000025808811]

ניתן לראות בגדול שהורדה של 10 אחוז מהמשקולות לא השפיעה בהרבה, ניתן לראות אבל שהפחתה של 0.3 העלתה את הדיוק מהמודל המקורי. בהפחתה של 60 אחוז מהמשקולות התחילה כבר ירידה בדיוק וב90 כמצופה איבדנו דיוק משמעותי.

אפשר לראות שבגרף המקורי הייתה תופעה של overfitting, הגרף של האימון הגיע ל100 אחוז דיוק בעוד גרף הטסט נשאר ללא שינוי. דבר זה מעיד בעיקר על יתר מורכבות של הרשת. קירבתי את הגרפים באזור ההתכנסות הסופי וזה נראה כך:





כפי שציפיתי לראות עבור 30 אחוז הפחתה (המודל שהגיע לדיקט הטוב ביותר) ניתן לראות את הגרפים של האימון ושל הטסט הכי קרובים מה שמעיד על למידה טובה ומדויקת יותר מהמודל המקורי.

לסיכום, שכבת dropout נועדה להפחתת מורכבות המודל על מנת להשיג תוצרי למידה טובים יותר ואכן ראינו שזה נכון בניסוי הקטן שערכנו.

שאלה 2.6

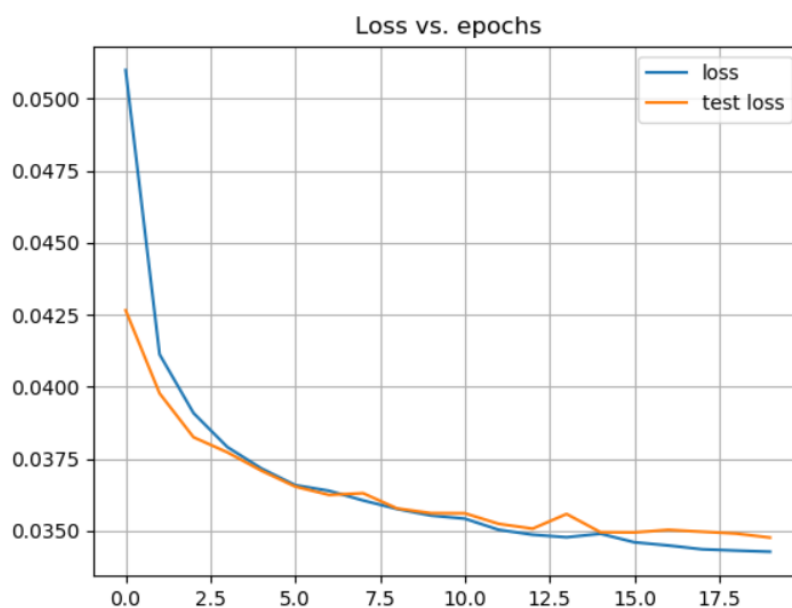
התחלתי בבניית המודל הנאיבי ביותר, שתי שכבות FC שמצמצמות מ 784 ל 2 וחזרה ל 784, על המודל הזה השוותי שני optimizers שאני מכיר שהם SGD שהשתמשתי במודלים הקודמים ו Adami שגם הוא מתבסס על רעיון דומה. נוכחתי לראות ש Adam השיג loss טוב יותר אז נשאיתי איתו.

עברתי לעבוד על מורכבות הרשת, הוספתי עוד כמה שכבות FC וראיתי שיש לכך השפעה לטובה על loss, בדקתי את ההשפעה של relu על loss וראיתי שהוספה של אקטיבציה כזאת עוזרת גם כן להפחית ב loss. בשלב זה החלטתי לשנות ארכיטקטורה ולבדוק את רשת הקובץ, הסתמכתי על מבנה דומה ל convent מהסעיף הקודם, לא ראיתי השפעה גדולה יותר מידי וסה"כ שני המודלים הגיעו לתוצאות דומות. החלטתי להישאר עם מודל ה FC בעיקר כי הייתי יותר בטוח באופן הבנייה שלו. 8 שכבות FC שמתכנסות למימד 2 באמצע וחוזרות בהדרגה ל 784 בפלט.

גרף ה loss :

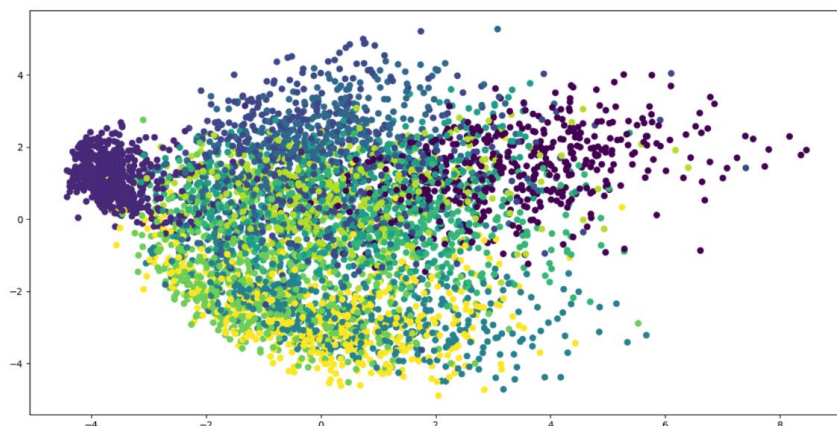
מתכנס לערך

של 0.034 על הטסט.



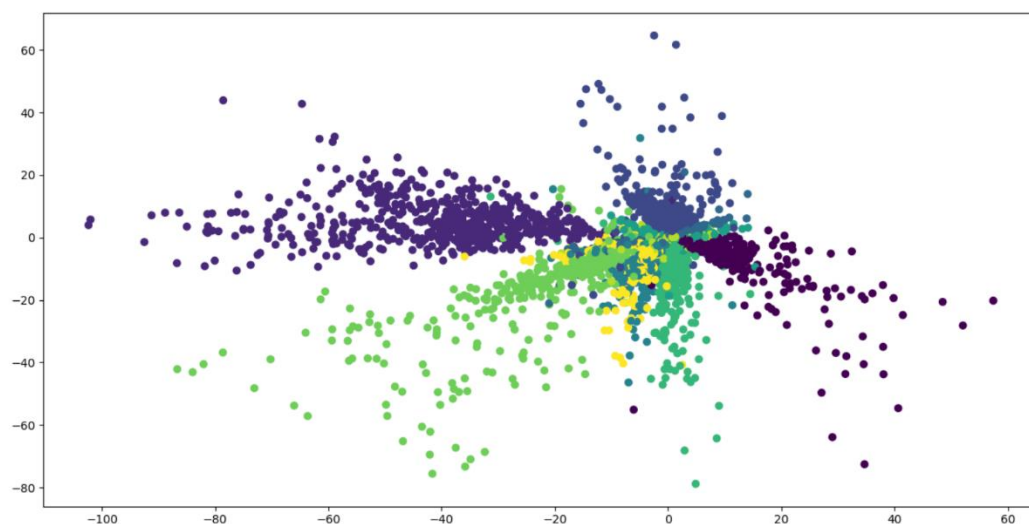
מודל הPCA:

כפי שהתבקש הרצתי את המודל לצמצום של 5000 דגימות להלן:



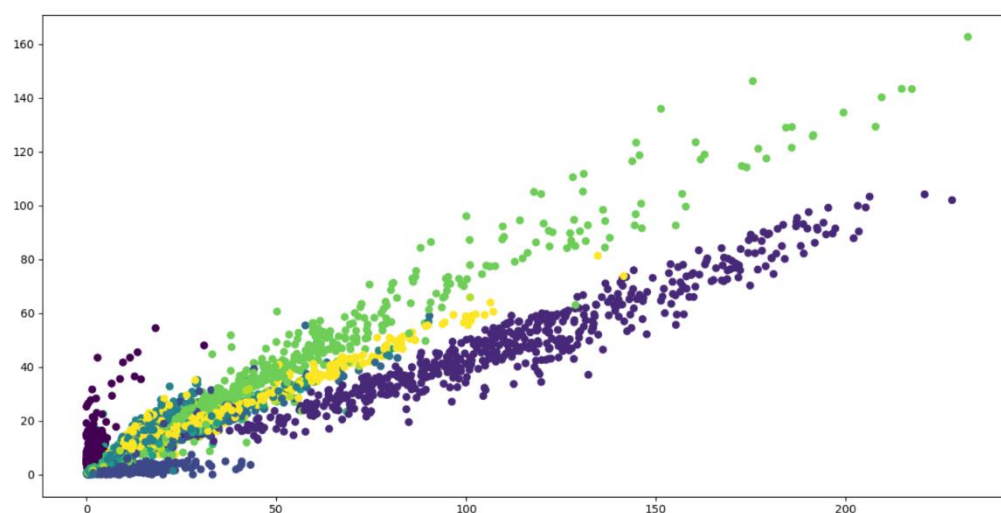
ניתן לראות חלוקה מסויימת לאזורים.

המודל שלי:



ניתן לראות במודל שלי חלוקה מסודרת יותר במרחב לפי התייגים, סה"כ כמצופה שוב מהיכולת של הרשת לבנות מודל מורכב יותר, למרות שהשתמשי בשכבות לינאריות אך ה relu מכניס גם גורם אי ליניאריות למשוואה.

רציתי לנסות להפעיל אקטיבציה על השכבה האמצעית של ה2, הוספתי relu וחזרתי על התהליך, ה loss הכללי היה זהה אך במרחב זה נראה עכשיו כך:

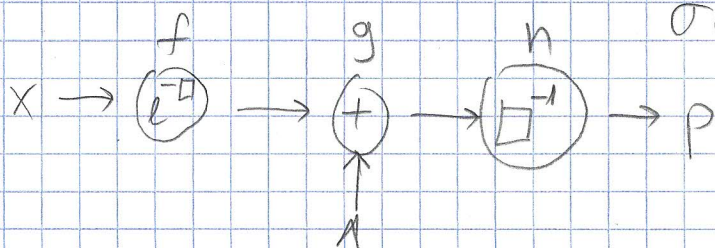


PART 1

1) נגזירה $f_i(0;t) = \max\{t, 0\}$

הוספנו הפונקציה המכונה "הוקס" אל מרחב ההפונקציות
שהיא משמשת כפונקציה אקטיבית. ה- \max המקסימום
הפונקציה הזו היא אקטיבית. ההוספה הזו היא אקטיבית
התוצאה היא הפונקציה הזו היא אקטיבית. ההוספה הזו היא אקטיבית.

2) $\sigma(x) = \frac{1}{1+e^{-x}}$



נחשב את הנגזרת $\frac{\partial p}{\partial x}$:

$$\frac{\partial p}{\partial x} = \frac{\partial p}{\partial h} \cdot \frac{\partial h}{\partial g} \cdot \frac{\partial g}{\partial f} \cdot \frac{\partial f}{\partial x} = \frac{1}{(1+e^{-x})^2} \cdot (-e^{-x}) = \frac{e^{-x}}{(1+e^{-x})^2}$$

הנגזרות של f היא $-e^{-x}$, הנגזרת של g היא 1 , הנגזרת של h היא 1 , והנגזרת של p היא 1 .

3) נגזרת של הפונקציה Log Convexity בהתאם לנגזרת $\frac{\partial l}{\partial u}$

1) $\frac{\partial l}{\partial u} = 2(p-y) \cdot m^T \rightarrow u=0$ כאשר $m=0$ ו- $\frac{\partial l}{\partial u}=0$

2) $\frac{\partial l}{\partial w_i} = \frac{\partial l}{\partial p} \cdot \frac{\partial p}{\partial m} \cdot \frac{\partial m}{\partial v} \cdot \frac{\partial v}{\partial w_i} \cdot \frac{\partial w_i}{\partial w_i} \rightarrow u=0$ כאשר $\frac{\partial l}{\partial w_i}=0$

כלומר זהו נגזרת של הפונקציה u, w_1, w_2 בהתאם לנגזרת $\frac{\partial l}{\partial u}$.
זהו נגזרת של הפונקציה u, w_1, w_2 בהתאם לנגזרת $\frac{\partial l}{\partial u}$.
כלומר זהו נגזרת של הפונקציה u, w_1, w_2 בהתאם לנגזרת $\frac{\partial l}{\partial u}$.