(67731) אופטימיזציה קמורה ושימושים | פרוייקט


בהצלחה

28 ביוני 2021

מידע אישי:

שם: עמית חן (308162502), דיוויד פיטס (313371080)
הערה - שנינו **תלמידי תואר ראשון** (מדעי המחשב - מדע הנתונים)

פירוש וניסוח השאלה:

ניסינו לכתוב את השאלה בצורה מעט שונה כדי להבין האם ניתן לאפטם את הבעיה (על ידי גזירה - להקל על זמן החישוב) בצורה נוחה יותר. ראינו כי:

$$Tr(SK) - log(|K|)$$

כאשר S מטריצה סימטרית וגם PSD, ואילו Kהיא מטריצת $K - band$ ו־PSD כפי שהוגדר בשאלה.
מכך ש־Sהיא מטריצת PSD, נשתמש בפירוק צ'ולסקי, $K = LL^T$.
(כאשר L מטריצה משולשית תחתונה ואיבירי האלסון שלה הם חיוביים ממש) . לכן:

$$Tr(SK) - log(|K|) = Tr(SLL^T) - log(|LL^T|)$$

כאשר נפתח את הביטוי הימני ונקבל (מפירוק דטרמיננת המכפלה למכפלת הדטרמיננטות, לאחר מכן השתמשנו בחישוב דטרמיננטה עבור מטריצה משולשית תחתונה ובתכונות של פונקציית ה־Log):

$$log\left(|LL^T|\right) = log\left(|L||L^T|\right) = log\left(\prod_{i=1}^{n} l_{i,i}\right) + log\left(\prod_{i=1}^{n} l_{i,i}\right) = 2\sum_{i=1}^{n} log(l_{i,i})$$

כאשר הנגזרת שלו לפי L היא:

$$\frac{log(|LL^T)}{\partial L} = diag(\frac{2}{L})$$

נשים לב שהאילוצים של מטריצה K($k-band$) נשמרים כיוון ש־ L משולשית תחתונה, ולכן מכפלה שלה בשחלוף של עצמה ייתן לנו
גם כן מטריצה משולשית תחתונה, כך שאם האלכסון שלה אי שלילי (ובמקרה של המכפלה שלה אכן אי־שלילי) אז מטריצה היא PSD ־ לכן האילוצים הנדרשים נשמרים.

כעת נחזור לחישוב הביטוי כולו , נגזור לפי L ונקבל:

$$\frac{\partial\left[Tr(SLL^T) - log(|LL^T|)\right]}{\partial L} = 2 * S * L - diag(\frac{2}{L})$$

...
כעת ננסה למצוא מכפלת מטריצות יעילה יותר עבור שיפור הביצוע של האלגוריתם שלנו.
נשים לב, לצורך הפשטות, עבור מטריצת מכפלה $B \in \mathbb{R}^{n \cdot n}$, כי $K = 2$ לכן מתקיים:

$$B_{1,1} = S_{1,1}L_1^0 + S_{1,2}L_1^{-1}$$

כאשר עבור L_1^0 מדובר באלכסון הראשי (0 של ה־ $k-band$) של המטריצה L ובעמודה הראשונה (1) שלה באותו האופן:

$$B_{2,1} = S_{2,1}L_1^0 + S_{2,2}L_1^{-1}$$

כאשר עבורL_1^{-1} מדובר באלכסון אחד מתחת לראשי (1־ של ה־ $k-band$) של המטריצה L ובעמודה הראשונה (1) שלה ובאופן כללי עבור השורות:

$$B_{i,1} = S_{i,1}L_1^0 + S_{i,2}L_1^{-1}$$

ולכן בכתיב כללי למטריצה כולה נקבל:

$$B_{i,j} = S_{i,j}L_j^0 + S_{i,j+1}L_j^{-1}$$

כעת, עבור תיאור של התהליך כאשר נרצה להתייחס לשורות של המטריצה B , **נשחלף** ונקבל:

$$B_1^T = S_1 \cdot L_1^0 + S_2 \cdot L_1^{-1}$$

ובאופן כללי עבור השורות:

$$B_j^T = S_j L_j^0 + S_{j+1} L_j^{-1}$$

ובאופן כללי על ידי סכימה:

$$B_j^T = \sum_{i=j}^{j+k} S_i L_j^{-(i-j)}$$

נשתמש בנוסחה זו באלגוריתם שלנו כדי לחשב ערכים בצורה יעילה.

זמן ריצה:

א. לכל עמודה במטריצה B הסופית שלנו נשים לב כי :

1. הכפלת k עמודות של המטריצה S בסקלר $L \in \mathbb{R}$, אז נעשה זאת באופן תאורטי ב־ $O(nk)$, אך במימוש של $numpy$ להבנתנו ניתן להשתמש בהיוריסטיקות כדי לייעל תהליך זה.

2. לאחר מכן, הוספה של כל החישובים הללו יחדיו תתבצע בזמן של $O(nk)$.

ב. נשרשר את כל התוצאות למטריצה (n וקטורים) אז זה יעשה בזמן $O(n)$־ לכן ביצוע **שלב א.** ו**שלב ב.** ייקח לנו $O(n^2k)$.

אלגוריתם ההתכנסות:

הערה: ההשערה ראשונית שלנו היא להשתמש באלגוריתם ניוטון־רפסון (לאחר ההתכנסות הראשונית , זאת לפי הרמזים ששמענו בכיתה מהמתרגל), זאת תוך שימוש בקמירות של f ,ואז בכך שההסיאן הוא מטריצה אי שלילית, ואז פירוק שלה

עם צ'ולסקי (כפי שראינו בכיתה) וכתוצאה מכך לחשב את ההסיאן בצורה קלה יחסית, אך חיפשנו דרכים יותר יצירתיות ומאתגרות להתמודד עם הבעיה (בנוסף, בתחום ה־ML למיטב הבנתנו האלגוריתם של ניוטון רפסון אינו בא לידי ביטוי באופן דומיננטי במסגרת הלימודים שלנו).

בחרנו להשתמש באלגוריתם GD עבור כל גודל של איטרציה. זאת מכיוון שהרגשנו כי עם הוספת היוריסטיקות אנו מוסיפים פוטנציאל רחב יותר "לשחק איתו" . בתור התחלה בחרנו לבצע GD עם הטלה כדי להתאים כל שלב באיטרציות עם מטריצה $K-band$ משולשית תחתונה. אחר כך בחרנו בהיוריסטיקות כפי שנפרט בהמשך הדו"ח.
נתאר את האלגוריתם של ההטלה עם GD:

1. נבצע GD"קלאסי" כפי שלמדנו בכיתה
2. נפכיל את הגרדיאנט שקיבלנו ב־GD במסיכה ($mask$) אשר מותאם למטריצה משולשית תחתונה עם ה־$k-band$ המתאים.
3. נמשיך באלגוריתם כרגיל.

קריטריון העצירה:

נרצה למצוא את הדרך האופטימלית לעצור את האלגוריתם כפונקציה של n,k כלומר $S(n,k)$, כאשר n זה המימד כמו מקודם, k מגיע כנתון מתוך ה־$k-band$.

קביעת ה־$Decay$:

בתור ניסיון ראשוני רצינו השתמש באיזשהו $cutoff$ על ידי שימוש בקבוע שנבחר, לדוגמא 0.5:

$$D_{lr,decay}(x) = lr \cdot (decay)^x$$

(נציין כי לבסוף לא השתמשנו בכך מכיוון שדרך זו מניחה שעבור סיבוכיות פרמטרים מסויימת, המרחב הנפרש על ידי כל S תלוי אך ורק בכמות הפרמטרים n,k ואינו תלוי בתכונות של S).
ברעיון זה רצינו לקבוע שה־$decay$ יהיה פונקציה של n,k.

ניתוח כמות הפרמטרים של המודל בהסתמך על n ועל k:

נתחיל בכך שנקבע את אחד מהמשתנים ונראה איך המשתנה האחר משפיע, באופן הבא:
עבור קיבוע של k:

$$f_k(n) = n * (2k+1)$$

עבור קיבוע של n:

$$f_n(k) = n * (2k+1)$$

כל עלייה של k מייצרת לנו $2n$ פרמטרים חדשים במודל, ואילו כל עלייה של n מייצרת לנו $(2k+1)$ פרמטרים חדשים במודל.
נשים לב ש־k חסום על ידי n ולכן העלייה בסיבוכיות של מושפעת יותר מעלייה של k מאשר עלייה של n.
לכן כאשר ננסה לאמוד את העלייה המשותפת של הפרמטרים שלנו נקבל את האומדן:

$$f(n,k) = n * k$$

על ידי מציאת התנהגות הפרמטרים נוכל למצוא חסם טוב יותר על גודל הגרדיאנט, מאחר והקריטריון לעצירה צריך להסתמך על הגרדיאנט:

$$f\left(n,k\right) = grad\,size$$

אין סיבה להאמין שערך גדול יותר של הגראדיאנט בכיוון אחד עשוי להשפיע על התכנסות מהירה יותר, לכן נסיק כי עלינו למצוא את ה־ $learning\ rate$ (להלן lr) באופן חכם (כי גודל הגרדיאנט משפיע לנו על הכיוון אליו נלך באלגוריתם, אך לא לא משפיע לנו על גודל הצעד, אלא ה־lr. ראינו דרכים להתמודדות עם בעיה זו ברשתות נוירונים לדוגמא (המקרה שם הרבה יותר מסובך) במקרה שלנו חשבנו על בחירת ה־lr המבוססת על גודל הגרדיאנט, באופן שיאפשר לנו התכנסות יציבה (כלומר לא להתבדר , "לקפץ מצד לצד בהתכנסות" או לא לרדת) בדרך למציאת המינימום של הפונקציה.

היינו מעדיפים להשתמש בפונקציה מהסוג $h' : \mathbb{R}^2 \to \mathbb{R}$ אך מכיוון שנו יודעים שלמרות שכמות הפרמטרים תהיה זהה ישנה השפעה שונה על התהליך של GD עבור הגדלה של n או של k אינה משפיעה באותה מידה על שינוי הערך של הפונקציה. זאת מכיוון שכאשר אנו נגדיל את n (עם k קבוע), אז ההטלה של ה־GD תאפס פרמטרים רבים יותר במטריצה.
לעומת זאת, ככל שנגדיל את k (עם n קבוע) אז מתבצעת הטלה של ה־GD על פחות פרמטרים במודל.
לדוגמא ־ עבור $k = n$ נקבל כי ההטלה של ה־GD אינה הולכת להשפיע לנו על המשלוש התחתון של המטריצה. לכן דיוק ההטלה גדל ככל ש־k גדל, ודיוק ההטלה קטן ככל ש־n קטן.
נציין שמההצגה שעשינו (ונעשה בהמשך) ענינו על השאלה המופיעה בקובץ "האם האלגוריתם מתחשב במימד של מטריצה S?"

נרמול גרדיאנט :

הבעיה שלנו היא כיצד להתאים את ה־lr לכל גודל של מטריצה שנוכל לקבל כקלט? הנחנו שהפתרון לבעיה היא שאנו חייבים לנרמל את הגרדיאנט.
חשבנו על דרכים לנרמל את הגרדיאנט:

א. חלוקת **בנורמת פרובינייוס** של המטריצה עליה למדנו בכיתה - לצערנו זה לא עבד כשבדקנו אמפירית
ב. חלוקת על ידי **הסכום** של כל האיברים - לצערנו זה לא עבד כשבדקנו אמפירית
ג. חלוקת עם הערך **המקסימלי** - עבד לא רע באופן אמפירי
ד. חלוקת בנורמה **אוקלידית** - עבד בצורה הטובה ביותר בבדיקה האמפירית
לכן בחרנו לנרמל את הגרדיאנט על שיטה קלאסית של נרמול אוקלידי (נורמה 2)

השפעת היציבות הנומרית של האלגוריתם:

כאשר הרצנו את האלגוריתם עבור ערכי n, k גדולים קיבלנו "בעיות" נומריות של התפוצצות הוקטנה משמעותיים הנבעו מחישוב מספרים קטנים\גדולים דיו עבור חישוב אלמנטרי של מחשב.
שמנו לב כי זה נבע מהפיכת האלכסון בחישוב הגרדיאנט ולכן ביצענו תיקון נומרי האומר שכאשר הערכים באלכסון קטנים מ-10^{-9} אז נציב במקום את הערך הנ"ל (10^{-9}) וזה ישמור על יציבות נומרית.

אתחול רנדומלי של מטריצה:

עבור אתחול רנדומלי L נבדוק את ערך הפנוקציה. אם הערך נמוך אז נניח שישנה התכנסות מהירה יותר לפתרון, אך אנו עשויים לטעות בהקשר זה.
לכן בחרנו לאתחל ולקבל 10 ערכים אפשריים עם lr יחסית גדול (0.5), ולבחור לבסוף את ה-lr שנותן לנו את התוצאה הטובה ביותר לאחר כמות איטרציות של האלגוריתם, שנתאר בעוד מספר רגעים.

אתחול כמות האיטרציות :

נאתחל את כמו האיטרציות בעזרת הפונקציה הבאה:

$$f(n,k) = \frac{300}{\sqrt{n * k}}$$

בחרנו את היחס הנ"ל מכיוון שלאחר **בחינה אמפירית** עבור גדלים שונים של כמות הפרמטרים במודל, ראינו שמודלים עם כמות פרמטרים n, k גדולה יותר הינם עמידים יותר ביחס לנקודת האתחול של האלגוריתם.
לכן בחרנו פונקציה שהינה יחס הפוך ככמות אתחול האיטרציות (מבדיקה אמפירית ראינו כי הכפלה ב-300 מסייעת לנו, וערכים גדולים יותר או קטנים יותר פחות טובים).
לדוגמא, עבור $n = 5, k = 3$ נקבל $\frac{300}{\sqrt{15}}$, כלומר בערך 75 איטרציות.
לעומת זאת עבור $n = 100, k = 25$ נקבל $\frac{300}{\sqrt{2500}} = \frac{300}{50}$, כלומר 6 איטרציות.

ניתוח סטטיסטי של השפעת כמות הפרמטרים במודל על חשיבות נקודת האתחול:

ביצענו את הסימולציות הבאות:

אתחלנו 1000 מטריצות רנדומליות. ועל כל אחת מהן הרצנו 50 איטרציות של GD . כלומר יצרנו מרחב מדגם של מטריצות שמאפשר לנו לבדוק מהן ההסתברויות עבור אתחול רנדומלי להתכנס באופן מהיר לפתרון הבעיה.

זאת על ידי השוואת הערכים ומציאת הערך הקטן ביותר של הפונקציה.

לכן קיבלנו ממוצע התחלתי עבור מטריצה בגודל $n = 5,\ k = 3$ (מתוך 1000 מטריצות) נקבל כי:

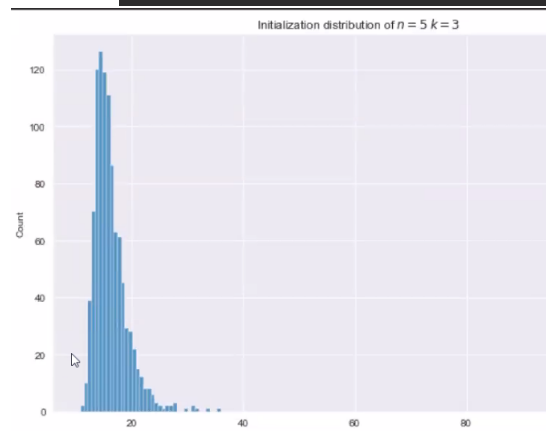
הממוצע: 16.152

סטיית התקן: 2.7

הערך המקסימלי:36.3

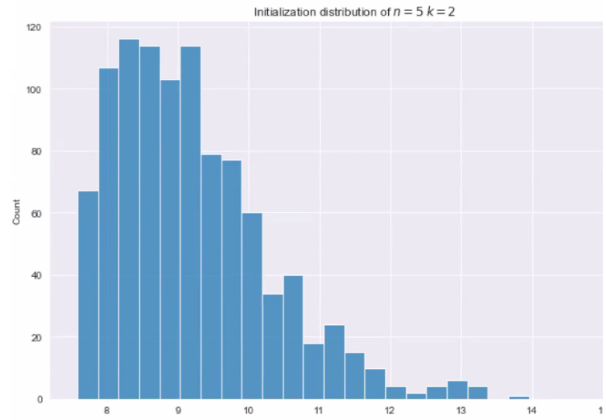
הערך המינימלי: 11.24

count	1000.000000
mean	16.152688
std	2.708276
min	11.243847
25%	14.311737
50%	15.645876
75%	17.327095
max	36.300576



דוגמא נוספת עבור מטריצה בגודל $n = 5,\ k = 2$:

	count	mean	std	min	25%	50%	75%	max
0	1000.0	9.119016	1.142528	7.558605	8.294791	8.802029	9.651575	16.129446



לעומת זאת עבור לכן קיבלנו ממוצע התחלתי עבור מטריצה בגודל $n = 10,\ k = 4$ נקבל כי:

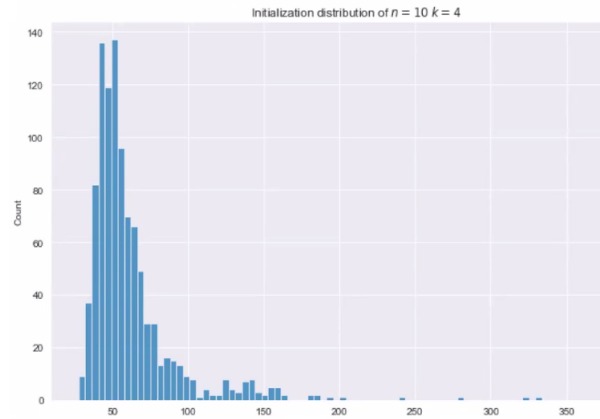
הממוצע: 66.272

סטיית התקן: 79.91 (סטיית תקן גבוהה)

הערך המקסימלי:2043.6

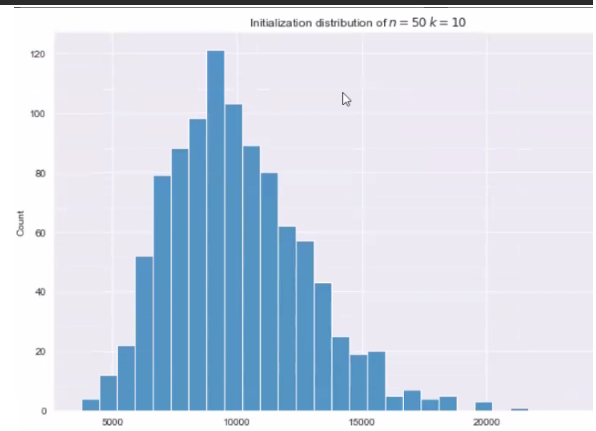
הערך המינימלי: 29.39

count	1000.000000
mean	66.272129
std	79.916141
min	29.393976
25%	44.751295
50%	52.795437
75%	65.921590
max	2043.677863



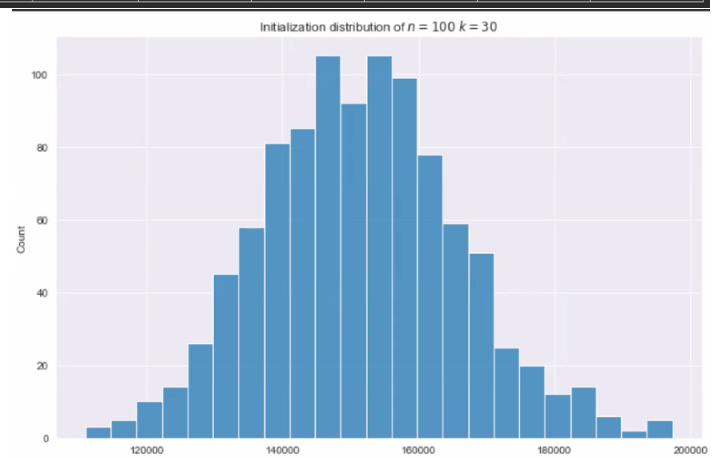
דוגמא נוספת עבור מטריצה בגודל $n = 50,\ k = 10$:

	count	mean	std	min	25%	50%	75%	max
0	1000.0	9932.582169	2832.903423	3961.993865	7919.925043	9567.772352	11682.170368	24788.982642



דוגמא נוספת עבור מטריצה בגודל $n = 100,\ k = 30$:

	count	mean	std	min	25%	50%	75%	max
0	1000.0	151508.175143	14427.996085	110921.936162	141359.102501	151420.215961	160957.591271	197351.633029



תובנות מהממצאים:

א. נראה שההיסטוגרמות הללו מצביעות על התפלגות נורמלית חרף ה־$outliers$ המעטים.
ב. הנטייה ל־$outliers$ ולהתפלגות נורמלית עם נטייה שמאלה (זנב ימני) הינה דומיננטית יותר בקרב ערכי פרמטרים נמוכים. לכן זה מצדיק את העובדה שבחרנו מספר איטרציות גדול יותר עבור ערכי פרמטרים נמוכים.
ג. כאשר אנו במרחק של sd מהממוצע אז ההסתברות להיות בטווח זה היא 68% ואילו עבור $2 \cdot sd$ היא 95%. לכן מההתפלגות הנורמלית שראינו אנו יכולים להניח כי באופן סביר שדגימה רנדומלית של נקודת אתחול עם פרמטרים גבוהים (גדולים מ־100) האתחול יהיה משתנה מקרי מהתפלגות נורמלית עם ממוצע ושונות המתאימים להיסטוגרמה. לכן ההסתברות להתחיל בנקודה תקינה הינה גבוהה מאוד.

לסיכום - ראינו כי ככל שכמות הפרמטרים עולה כך המודל עמיד יותר ביחס לנקודת האתחול, לעומת זאת עבור פרמטרים נמוכים המודל פחות רובסטי וכולל $outliers$ רבים, לכן ישנה סבירות לא מבוטלת להתחיל בנקודת אתחול "פחות טובה". ממצאים אלו מצדיקים את ההצגה של היחס בין כמות הפרמטרים לגודל האתחול שהצגנו תחילה (קרי $f(n,k) = \frac{300}{\sqrt{n*k}}$).

שיטות אפשריות ל־lr דינאמי:

חשבנו על 2 אפשרויות לבחירת ה־lr
1. lr ציקלי - נבחר חסם עליון וחסם תחתון ונעבור בינהם באופן מעגלי עבור t צעדים, ולאחר מכן נצמצם את הטווח הזה פי 2 (כלומר נחתוך את החציון העליון של הטווח) זאת כדי להקטין את מספר אפשרויות הבחירה של ה־lr, ונחזור חלילה.
2. סוג של "$Line\ Search$"כך שעבור t צעדים נבחר עשרה lr ונבדוק אותם עבור 20 צעדים, ולבסוף נבחר את ה־lr שנותן לנו את התוצאה האופטימלית, כלומר מקרב אותנו למינימום של הפונקציה. כאשר:
א. ממוצע נמוך , כאשר ניקח ממוצע של 3 הצעדים האחרונים. זאת על מנת להתמודד עם אי־יציבות שיכולה להתקיים באלגוריתם (וכי באופן כללי מומלץ להשתמש באלמנטים סטטיסטיים כדי לקבל תובנות על אלגוריתם עם המון איטרציות).
הסבר נוסף לכך הוא כדי להימנע מקפיצה בין צדדים של הצורה הקמורה של הפונקציה (כלומר זה מעיד על lr גדול מדי) אז נצמצם את ה־lr.
בחרנו באופציה השנייה מכיוון שלהרגשתנו lr ציקלי פחות מתאים לבעיות קמורות ויותר מתאים לבעיות של למידה עמוקה, כמו ברשתות נוירונים.
אולם, כאשר בחרנו גישה זו מצאנו בעייתיות מסויימת: שמנו לב לדפוס בו כאשר אנו נותנים לאלגוריתם לבחור בין p אפשרויות של lr שונים (כלומר מרחב אפשרויות רב), אז האלגוריתם כמעט תמיד בחר את האופציה המינימלית של ה־lr מבין האפשרויות שהוצעו לו. לכן חיפשנו דרך לשנות גישה זו, כלומר לנצל את הכוח של שיטת חקירת ה־lr מאפשרת.
לכן לאחר כל t צעדים עם אותו ה־lr , בדקנו מכל האפשרויות איזה lr נבחר, ולאחר מכן התאמנו את הטווח המינימלי והמקסימלי של חיפוש ה־lr שהצגנו קודם, על פי ה־lr שנבחר בצעד הקודם, כלומר בחרנו לחפש בטווח של 0.5 עד 1.5מהערך של ה־lr שנבחר:

$$l_{opt}^{(i+1)} \in [0.5 \cdot l_{opt}^{(i)}, 1.5 \cdot l_{opt}^{(i)}]$$

שיטה זו אפשרה לנו לתת לאלגוריתם ה־GD את מירב האפשרויות לבחור את ה־lr כדי לאפטם את מציאת הפתרון לבעיה. ואכן, בתוצאות ראינו כי ה־lr שנבחרים אינם תמיד הקטנים ביותר או הגדולים ביותר כפי שראינו בהתחלה, והבחירה הייתה מגוונת יותר.

ביצוע סימולציות להשגת פונקציה המספקת תנאי עצירה כתלות בפרמטרים:

נרצה למצוא סף על גודל הדרדיאנט כדי למצוא תנאי עצירה לאלגוריתם,נצפה כי עבור כל כמות פרמטרים שונה, יהיה גם סף מתאים לגודל הגרדיאנט, שמייצג עבורנו מתי הגענו לנקודה אשר תניב מטריצה K אשר נמצאת במרחק "מספק" מהמטריצה האופטימלית.

לכן בשביל למצוא את הספים נרצה לסנתז דאטה, כלומר להגריל מטריצה S שהיא PSD וסימטרית, אך כרגע אין לנו מטריצה אופטימלית אשר נוכל להשוות אליה. לכן, נרצה למצוא מטריצה אופטימלית עבור S נעשה זאת על ידי למידת מטריצה K ונבטא את תנאי העצירה שלנו כתלות בגודל הגרדינאט בלבד (תנאי העצירה בשלב זה יהיה גדול יותר ככל שכמות הפרמטרים תגדל וזה נעשה על ידי תצפית אמפירית של גדלי הגרדיאנט בדוגמאות שהובאו לנו). כעת נניח כי קיימת מטריצת $K^\star$ אופטימלית עבור מטריצה S שהגרלנו. כעת נגריל p מטריצות L ועבור כל אחת כזו אז נריץ את אלגוריתם ה־GD עד שנגיע להתכנסות על פי מטריקת פרוביניוס שניתנה לנו עם שגיאת האפסילון הרצויה (10^{-3}), במילים אחרות נצליח לאסוף מידע על גודל הגרדיאנט השקול לתנאי המספיק לעצירה אשר תניב לנו מטריצה K שתהיה במרחק הנדרש מ־$K^\star$.

נוכל לאסוף מידע סטטיסטי ולבנות בעזרתו מודל שיצליח לחזות את גודל הגרדיאנט על סמך כמות הפרמטרים. נעשה זאת עבור הגרלות רבות של מטריצות ועבור ערכי n, k שונים, כך שנוכל "למצוא פונקצייה" ($h : \mathbb{R} \to \mathbb{R}$) שמתארת את הקשר בין כמות הפרמטרים ($n \cdot (2k+1)$ שתייצג את ציר ה־x) לבין גודל הגרדיאנט שאנו מחפשים (שיהיה ציר ה־y של הפונקציה), תיארנו בעבר כי הפונקציה האופטימלית עבור מקרה כזה תהיה מ־$\mathbb{R}^2$ ל־$\mathbb{R}$, אך בדיקה סטטיסטית עבור טווח דו מימדי ידרוש מאיתנו סנתוז מטריצות בזמן ריצה שאין ביכולתנו לממש כרגע, ולכן נסתפק בפונקציה פשוטה יותר, כלומר מ־ $\mathbb{R}$ ל־$\mathbb{R}$. כלומר הפונצקיה תהיה:

$$f(n, k) = (2k + 1) \cdot n$$

אך אנו נתייחס אליה כאילו הינה מ־ $\mathbb{R}$ ל־$\mathbb{R}$, כלומר נתעלם מהעבודה כי אותם כמות הפרמטרים יכולה לנבוע מ־2 ערכי n, k שונים.

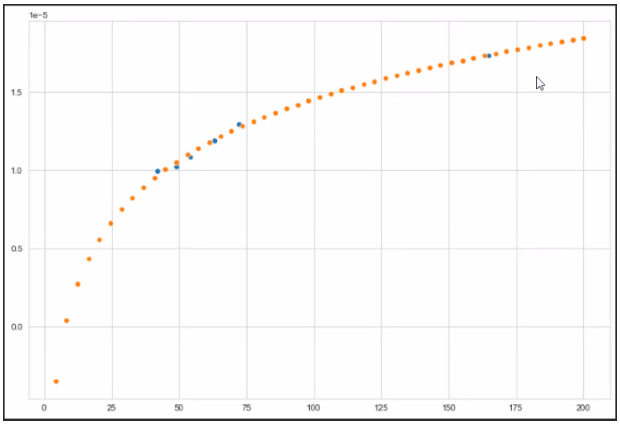
לכן כעת נשאלת השאלה כיצד נלמד את הפונקציה הזו? אספנו את כל הערכים מסימולציות של מטריצות שונות במימדים שונים. כעת נבצע $scatter - plot$ עבור כל הדאטה שאספנו ונתבונן בתופעות המופיעות בגרף שקיבלנו. על פי כך נחליט מהי הפונקציה המתאימה (פונקציה לינארית, ריבועית, מעריכית וכו'..) . לאחר מכן נלמד את הפרמטרים המתאימים של הפונקציה , ונקבל פונקציה המתאימה בין פרמטרים לתנאי העצירה שתלוי בגודל הגרדיאנט. כעת נוכל להשתמש בתנאי עצירה זה באופן גלובלי עבור כל מטריצה שנקבל. במילים אחרות , לכן בהינתן המידע על n, k נבדוק מה הערך שיוחזר לנו מהפונקצייה h (כלומר גודל הגרדיאנט), ונעצור את האלגוריתם אם נגיע לגודל הגרדיאנט הזה, שמהווה תנאי עצירה שיבטיח לנו קירבה ל־K האופטימלי.

מפאת פערים חישוביים לא הצלחנו ל־"סנתז" מספיק דאטה אך ניסינו ליצור סימולציה שתעיד על גודל הגרדיאנט, המהווה כזכור תנאי עצירה, כלתות בגודל המטריצה S (ציר ה־ x הוא כמות האיטרציות של האלגוריתם, ואילו ציר ה־y זו השגיאה מהאופטימלי).

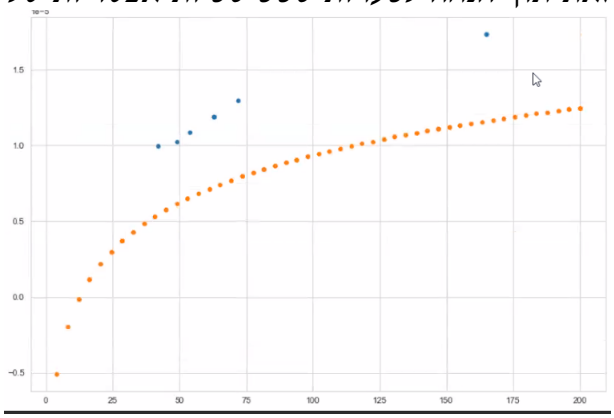
בתמונה הראשונה ניתן לראות כי הנקודות מואתמול להתפלגות לוגריתמית, זאת גם בשילוב של אמונה פריורית שלנו שכך הדאטה מתפלג.

הפונקציה היא:

$$f(x) = 5.63^{-6} * log(x) - 1.142^{-5}$$



עקב כך שייתכן אי דיוק בעקבות הסימולציות בחרנו לסרטט קו שמרני ביחס לקו שהוצג בתמונה הקודמת בו הכפלנו את שיפוע הגרף ב־0.8 , זאת ניתן לראות בתמונה הבאה, זאת על מנת להשיג חסם תחתון שנותן לנו הבטחה על גודל הגרדיאנט, זאת תוך הנחה לטעויות סטטיסטיות אפשריות של המודל.



עבור התמונה הבאה, פיזרנו 3 נקודות שעבורן במהלך האימונים מצאנו כי ה־lr ($step\ size$) הזה הביא לתוצאות הטובות ביותר:

הנקודות הן?

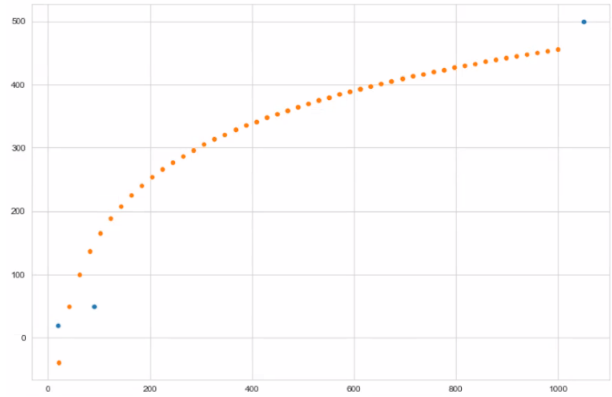
א. $(20, 20)$

ב. $(90, 50)$

ג. $(1050, 500)$

הפונקציה הינה:

$$f(x) = 127.23 * log(x) - 422.938$$



בשני המקרים נבחרה פונציה לוגריתמית בכדי שהיחס בין n ו־k גדולים מאוד עדיין ימשיך באותה המגמה אך לא "יתפוצץ" ויקבל ערכים גדולים מאוד.