**Section 1: Linear regression implementation**

(Q1)

הפונקציה היא פונקציית מקרים.

נתייחס לנגזרת שלה לפי b לפי המקרים:

* עבור המקרה , הטעות היא ריבועית ביחס לb והפונקציה היא גזירה ביחס לb. אין נקודות אי גזירות בתחום זה ולכן הגדרת הנגזרת והsubderivative מתלכדות.
* עבור המקרה , הטעות היא מוגדרת באמצעות הזזה והכפלה בסקלר של פונקציית הערך המוחלט- . פונקציה זו היא גזירה ביחס לb בכל מקום מלבד באופן פוטנציאלי בנקודה שבה . עם זאת, מאחר שאנו בתחום שבו , הפונקציה גזירה באזורים אלו גם כן, ולכן הגדרת הנגזרת והsubderivative מתלכדות.
* עבור המקרים בהם , כלומר כאשר
* הפונקציה עדיין גזירה ביחס לb בנקודות אלו כיוון שהנגזרות מימין מתלכדות עם הנגזרות משמאל.

סה״כ מאחר והפונקציה גזירה ביחס לb על כל התחום נוכל לחשב את הנגזרות החלקיות בנפרד:

נחשב כל חלק בנפרד:

1. :
2. Otherwise :

נקבל :

(Q2)

נניח שיש לנו m זוגות של דוגמאות ותיוגים, כאשר כל דוגמה ממימד d כלומר :

נסמן את האלמנטים של דוגמה מסוימת, באופן הבא:

אזי נגדיר את המטריצה ונגדיר את וקטור העמודה .

עבור וקטור נגדיר פונקציית אינדיקטור באופן הבא

: .

עבור וקטור נגדיר פונקציית אינדיקטור באופן הבא

: , כאשר הוא האלמנט במיקום הi בוקטור z .

נשים לב כי

בנוסף, עבור וקטור נגדיר פונקציית אינדיקטור באופן הבא

: , כאשר הוא האלמנט במיקום הi בוקטור z .

אזי נוכל לבטא את באופן הבא:

נבדוק שהנוסחה חוקית:

* = היא מכפלה חוקית ונותנת לנו וקטור עמודה בעל m מימדים, לכן גם הסכום () כיוון שכל הוקטורים שייכים ל.
* הקלט ל הוא ואכן והפלט הוא גם כן וקטור
* מכפלת הדמרד (איבר – איבר) בין ובין

ולידית כי שניהם מאותו מימד והפלט הוא גם כן ב

* ולכן חוקי להכפיל בין המטריצה לווקטור

ונקבל פלט של וקטור בגודל המצופה לוקטור גרדיאנט כיוון שאנו גוזרים לפי .

* סה״כ הביטוי מחזיר לנו כפי שאנחנו רוצים.
* הביטוי הוא ולידי כיוון שראינו כי ולכן חישוב ההפרש בין 2 בוקטורים הוא לידי ומתקבל וקטור ששייך ל .
* לפי איך שהגדרנו את sign לוקטורים ממימד m, אזי הביטוי הוא ולידי ומחזיר וקטור ששייך ל .
* ולידי כיוון שאנו מבצעים מכפלה איבר-איבר בין שני וקטורים ממימד m.

כמו קודם, גם המכפלה חוקית כיוון שכופלים בוקטור מימד m. נקבל פלט של וקטור בגודל המצופה לוקטור גרדיאנט כיוון שאנו גוזרים לפי .

* סה״כ נקבל מהסכום של 2 הביטויים, 2 וקטורים ממימד d, לכן פעולת החיבור הזו היא אכן ולידית ואנחנו מקבלים וקטור גרדיאנט ממימד d כמצופה.

כעת נפתח נוסחה עבור

נבדוק שהנוסחה חוקית:

* = היא מכפלה חוקית ונותנת לנו וקטור עמודה בעל m מימדים, לכן גם הסכום () כיוון שכל הוקטורים שייכים ל.
* הקלט ל הוא ואכן והפלט הוא גם כן וקטור
* ולידי מאחר ומדובר במכפלה פנימית בין 2 וקטורים מימד m ובסופו של דבר הפלט מהמכפלה הזו הוא מספר כפי שהיינו מצפים.
* גם הביטוי הוא מספר , כמצופה.
* הביטוי הוא ולידי כיוון שראינו כי ולכן חישוב ההפרש בין 2 בוקטורים הוא לידי ומתקבל וקטור ששייך ל .
* לפי איך שהגדרנו את sign לוקטורים ממימד m, אזי הביטוי הוא ולידי ומחזיר וקטור ששייך ל .
* ולידי מאחר ומדובר במכפלה פנימית בין 2 וקטורים מימד m ובסופו של דבר הפלט מהמכפלה הזו הוא מספר כפי שהיינו מצפים.
* סה״כ הביטוי מחזיר מספר ממשי.
* לכן הסכום בין שני הביטויים של הנגזרות לפי b חוקי ומחזיר לנו מספר ממשי , כמצופה.

(Q3)

Algorithm: CalculateHuberDelta

Input: dataset X (features), y (labels)

Output: delta (float)

1. Fit a linear model to the dataset (X, y).

2. Calculate the absolute residuals from the model's predictions.

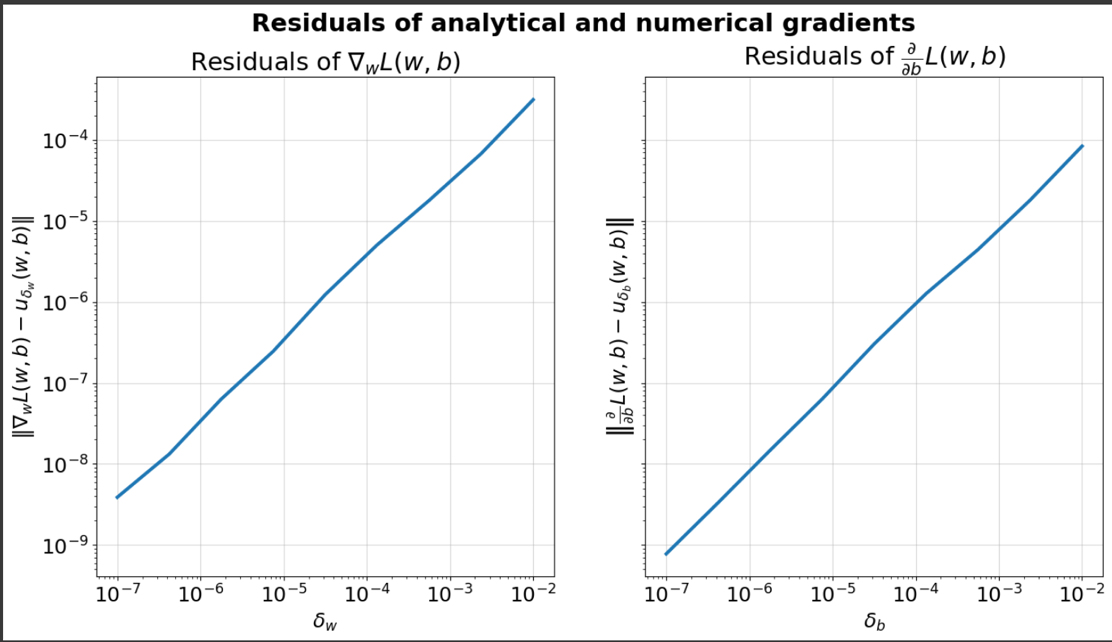
3. Perform statistical analysis on the residuals:

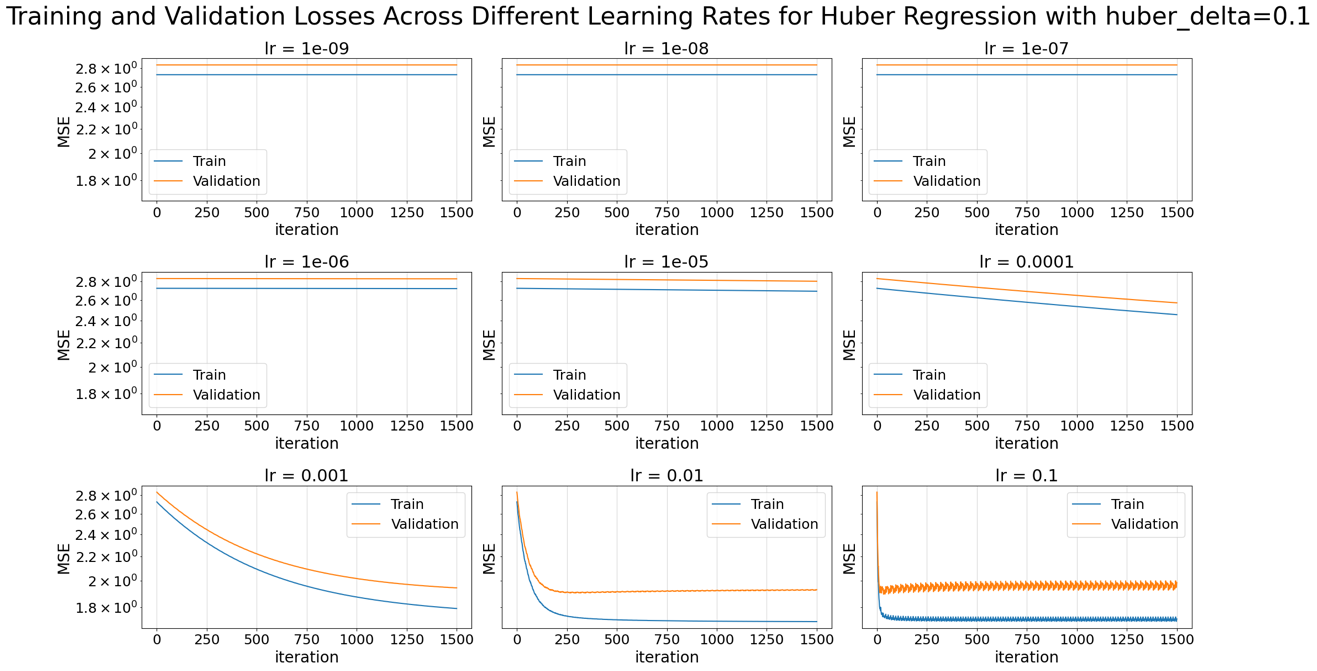
a. Compute the 25th percentile (Q1) and the 75th percentile (Q3) of residuals.

b. Compute the Interquartile Range (IQR) as Q3 - Q1.

4. Define delta as Q3 + 1.5 \* IQR, which is the standard heuristic for the upper boundary of normal variation in a dataset.

Return delta

(Q4 )

 (Q5)

השתמשנו בhuber\_delta= ע״פ האלגוריתם שייצרנו בסעיף 3.

ניתן להצדיק שרואים בגרפים כך:

\*עבור קצבי למידה נמוכים (1e-09-1e-5), ניתן לראות שבקושי יש ירידה בloss הן בtrain והן בvalidation מאחר וככל הנראה גודל הצעד בgradient קטן מדי ולכן ההתכנסות איטית מאוד ולא קורית ב1500 צעדים.

\* כאשר קצב הלמידה עולה לlr=0.0001 אנו רואים כי כן יש ירידה קטנה בloss, הן בvalidation והן בtrain, אך עדיין הירידה הזו קטנה ביחס לloss שמגיעים אליו בקצבי למידה גבוהים יותר. עבור קצב למידה הזה, אנו אכן מתקרבים לכיוון המינימום אך כמות הצעדים לא מספיקה עדין להתכנסות.

\* עבור קצבי הלמידה 0.001,0.01 אנו רואים כבר שחלה ירידה משמעותית בloss במהלך האימון והם מגיעים לערכי loss יחסית דומים בסוף האימון עבור שני הקצבים. עם זאת, הירידה עבור הקצב 0.001 היא מתונה יותר, בהשוואה ל0.01 , מאחר שגודל הצעד קטן יותר ולכן אנחנו מתקדמים לעבר המינימום בצורה איטית יותר.

\* עבור קצב הלמידה 0.1 , ההתכנסות היא מהירה מאד אבל יש הרבה יותר תנודות בערך הloss בתהליך האימון וגם בקבוצת ולידיה. התנהגות זו מצביעה על כך שקצב למידה זה גדול מדי ולכן האלגוריתם מפספס את המינימום בכל איטרציה.

\* עוד נשים לב, שערך הloss של קבוצת האימון ושל קבוצת הוולידציה דומה במהלך כל תהליך האימון של המסווג, ושווקטור w כלשהו שאנו מוצאים במהלך תהליך האימון שמקטין את הloss על הtrain, מקטין זאת את הloss על הvalidation. התנהגות זו היא התנהגות טובה, שמראה לנו שהמסווג שלנו אכן לומד במהלך תהליך האימון ויודע לבצע הכללה על דאטה שלא ראה (validation). לא נראה שהגענו עדיין לנק׳ overfit במהלך תהליך האימון כיוון שלא ראינו אינדקציה לכך שהtrain ממשיך לרדת אבל הloss של הvalidation כבר מתחיל לעלות. כמו כן, ברור שעבור קצב הלמידה 0.01 הייתה התכנסות מהירה יחסית למינימום כלשהו לאחר מס׳ נמוך של צעדים, ואימון נוסף בקושי מקטין את הloss על הtrain והvalidation.

**לדעתנו , קצב הלמידה האופטימלי הוא lr=0.01** כיוון שהוא מתכנס למינימום, ללא תנודות כמעט ובצורה מהירה יותר מאשר הקצב 0.001. בנוסף, הוא מקטין את הloss גם על הtrain וגם על הvalidation והloss שקיבלנו על קבוצת הולידציה היה הנמוך ביותר עבור קבוצה זו.

הגדלה של מספר הצעדים עבור lr זה אינה הכרחית כיון שניתן לראות שהאלגוריתם הגיע כבר לנק׳ מינימום כבר לאחר מס׳ מועט של צעדים, ואימון נוסף כבר בקושי הקטין את הloss.

Q6))

הrobustness של פונקציית הhuber loss מנוצלת בצורה טובה היותר כאשר הדאטה מכיל outliers , כאשר הדאטה מגיע מהתפלגות עם זנב ארוך (יש הסתברות נמוכה לקבל ערכים מאוד רחוקים מהערכים שבדר״כ מקבלים) או כאשר ההנחה שלנו שהרעש מתפלגת נורמלית נשברת. במקרים כאלו, רגרסיה מסוג OLS (בעיית הריבועים הפחותים הרגילה) עובדת פחות טוב כיוון שהיא רגישה מאוד למדידות חריגות. מהגדרת בעיית הריבועים הפחותים באמצעות שגיאה ריבועית, ערכי מדידות חריגים שיש להם שגיאה גדולה יותר מהשגיאה הממוצעת, משפיעים הרבה יותר על גודל הloss מאשר המדידות האחרות ובכך גורמים לקו הרגרסיה להתאים את עצמו הרבה יותר לרעש. המאפיינים של פונקציית הhuber מסייעים לה להתמודד עם בעיית הרעש- פונקציית loss זו מתנהגת כמו שגיאה ריבועית כאשר הresiduals קטנים ומתנהגת כמו שגיאה לינארית כאשר הresiduals גדולים מסף מסויים. הסף, delta, הוא היפר-פרמטר שניתן לכוונון ולמעשה מאפשר לנו להגדיל/להקטין את העמידות (robustness) של פונקציית הloss מפני חריגות. ערכים נמוכים יותר של delta יהיו פחות רגישים לoutliers בהשוואה לערכים גבוהים יותר, עם זאת אנחנו לא רוצים להתייחס לכל הדאטה כ״רעש״ או לגרום כבר לunderfitting ובעיות אופטימיזציה אלא אנו רוצים להמשיך לנצל את היתרונות של בעיית הריבועים הפחותים הרגילה.

למעשה, פונקציית הhuber היא סוג של שילוב בין השגיאה הריבועית לשגיאה האבסולוטית, כאשר היא מאפשרת לנצל את היתרונות של שתי השיטות- השגיאה האבסולוטית אינה גזירה ב0 וקשה לאופטימיזציה, בעוד שפונק׳ הhuber מתנהגת סביב 0 כמו שגיאה ריבועית ולכן גזירה. עם זאת, עבור ערכי שגיאה גדולים, פונק׳ הhuber פחות רגישה לרעש כיוון שנותנת פחות חשיבות לרעש ואנומליות (שם השגיאה לינארית).

(Q7)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Valid MSE**  (cross validated)  Mean (std) | **Train MSE** (cross validated)  Mean (std) | **Train Huber Loss**  (cross validated)  Mean (std) | **Section** | **Model** |
| 5.29 (+/- 1.051) | 5.27 (+/- 0.26) | 2.29(+/- 0.1) | 2 | Dummy |

(Q8)

הערכים שבחרנו לבדוק עבור ה learning rateהם 31 ערכים ברווחי log בטווח

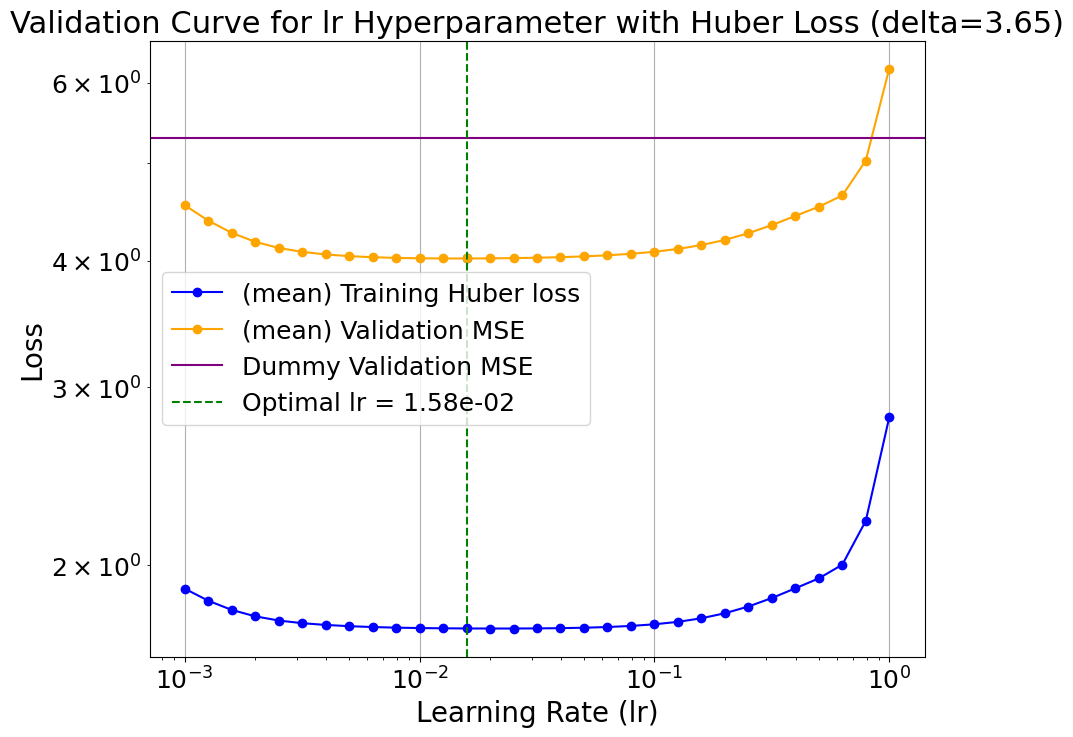
)כך יצא שההפרש בין חזקה לחזקה הוא 0.1)

בחרנו בטווח זה כיוון שראינו בשאלה 5 שהאלגוריתם שלנו מתכנס עבור learning rates בין אזור ושבאזור יש קצת רעש אך עדיין היתה התכנסות. לכן החלטנו להרחיב את טווח החיפוש על טווח הlearning rates הזה כדי לאפטם את הlearning rate כך שנקבל שגיאת ולידציה הכי קטנה.

ע״פ ההוראות, אימנו עבור כל learning rate מודל רגרסיה המאומן על huber\_loss (השתמשנו בcross validation עם חלוקה ל5 קבוצות כך שלמעשה לכל learning rate אומנו 5 מסווגים).

עבור כל learning rate חישבנו את ממוצע שגיאת האימון וממוצע שגיאת המבחן (ממוצע על חמשת הסגמנטים מהcross validation).

להלן הגרף שקיבלנו:



כפי שניתן לראות, הן שגיאת האימון והן שגיאת המבחן מתנהגות באופן הבא: ככל שהlr עולה השגיאה הולכת ויורדת ולאחר מכן עבור lr גדולים מדי השגיאה מתחילה לעלות בחזרה. ההתנהגות הזו צפויה , כי כפי שראינו קודם, עבור lr גדולים מדי , אלגוריתם הsgd כבר מתחיל להתבדר ולכן לא מקטין את השגיאה. עבור lr קטנים, ההתכנסות של האלגוריתם איטית יותר ולכן גודל הloss שמגיעים אליו במספר קבוע של צעדים הוא עדין גדול לעומת ה שאפשר להגיע עם lr גדול יותר.

נשים לב כי אנו מודדים את השגיאה באימון והשגיאה במבחן באמצעות מטריקות שונות , וההפרש בין שגיאת האימון לשגיאת המבחן נובע בין השאר גם מההבדל בחישוב (עבור הhuber loss , לא מעלים בריבוע שגיאות גדולות מסף דלתא).

הlearning rate האופטימלי שמצאנו באימון הוא:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Valid MSE**  (cross validated) | **Train MSE** (cross validated) | **Train Huber Loss**  (cross validated) | **Section** | **Model** |
| 5.29 (+/- 1.051) | 5.27 (+/- 0.26) | 2.29(+/- 0.1) | 2 | Dummy |
| 4.2 (+/- 1) |  | 1.72(+/- 0.1) | 2 | Linear |

(Q9)

עבור המודל הdummy שחוזה תמיד ערך אחד והוא הy הממוצע, לנרמול אין שום אפקט. זאת מכיוון שמודל זה בכלל לא מתייחס מה הם הפיצ׳רים של דוגמה ספציפית בעת חיזוי, אלא הוא תמיד חוזה את אותו ערך הy הממוצע (ערך יחיד).

עבור המודל הלינארי עם huber-loss –

ברמה התאורטית-

שיטות הנרמול אליהן נחשפנו בקורס- min-max וstandardization הן טרנספורמציות לינאריות:

מאחר ושיטות הנרמול שלנו הן טרנספורמציות לינאריות, נראה כעת שלכל מודל רגרסיה המאופיין ע״י המשוואה ניתן למצוא מסווג אחר המאופיין ע״י כך ש.

נניח שיש לנו דאטהסט ואנו מבצעים טרנספורמציה לינארית לכל אחד מהפיצ׳רים כך ש . כלומר עבור כל אנו מקבלים .

נניח ש אזי נגדיר

ונקבל:

מההוכחה אנו מקבלים שקיים קשר חד-חד ערכי ועל בין הפרמטרים w,b עבור x לפני נרמול לפרמטרים w’,b’ עבור x לאחר נרמול. הסט של ההיפר-מישורים האפשריים לפני נרמול ואחרי נרמול הוא למעשה אותו סט. מכאן, שהערך של הtraining loss, ובמיוחד הערך האופטימלי של הtraining loss ישאר האותו הדבר ברמה התאורטית במהלך תהליך אימון (כלומר תמיד נוכל להגיע לאותו ערך, לא לאותו arg min ).

בתהליך האימון שלנו אנו מחפשים מינימום עבור פונ׳ מטרה קמורה שהיא הhuber-loss. הloss הוא בסופו של דבר פונקציה של הw,b . כפי שראינו, הנרמול לא אמור להשפיע על ערך הloss שאנו יכולים להשיג כיוון שכל ערך loss שניתן להשיג עם w,b שמוצאים לפני נרמול אפשר למצוא w’b’ שיתנו לנו את אותו ערך loss ולהפך.

עם זאת, כפי שראינו בהרצאה, ברמה הפרקטית , הביצועים יכולים להיפגע כאשר לא מנרמלים את הדאטה. בנועבור דאטהסט מנורמל/לא מנורמל נקבל גרדיאנט שונה (עבור נק׳ w כלשהי שנחשב לה את הגרדיאנט שכן במקרה של הhuberloss הגרדיאנט תלוי גם בערך של הפיצ׳רים ולא קבוע.

עבור פיצ׳רים שהסקלה שלהם גדולה יחסית, נצטרך learning rate נמוך יותר (כיוון שהנגזרת החלקית לפי המשקולת המתאימה לפיצ׳ר תהיה גדולה יותר) מאשר פיצ׳רים שהסקלה שלהם יחסית קטנה . כתוצאה מכך, נצטרך לבחור lr קטן מספיק לפיצ׳ר עם הסקאלה הכי גדולה. יכול להיווצר מצב שנבחר lr נמוך מדי עבור הפיצ׳רים עם הסקלה הקטנה וההתכנסות תהיה הרבה יותר איטית, כלומר נצטרך יותר איטרציות כדי להתכנס .