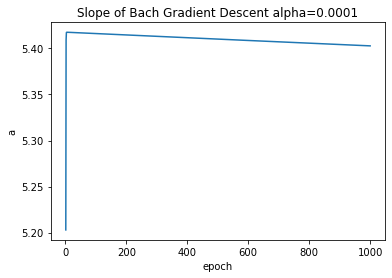
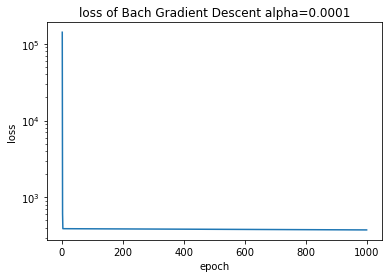
**Batch Gradient Descent (BGD)**

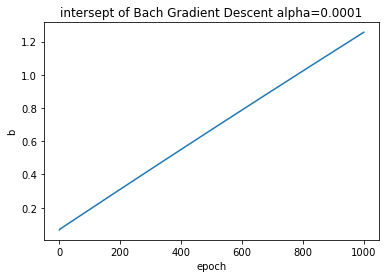
שיטה זאת היא השיטה האיטית ביותר והמדויקת ביותר. נעדיף להשתמש בה כאשר יש לנו DATA לא גדולה במיוחד. עבור DATA גדולה העומס על הRAM יהיה גבוה מידי והמחשב לא יוכל להריץ אותה או פחות לא בזמן סביר. בתחילת כל איטרציה נלקחים צעדים גדולים לכיוון נקודות המינימום הגלובלית מכיוון שיש שיפוע גדול. לאחר מכן ככל שהשיפוע קטן קצב ההתקדמות קטן עד שמגיעים בקרוב לנקודה המינימלית אותה חיפשנו. בשיטה זו בכל איטרציה יהיה מעבר על כל הDATA במקרה שלנו 2048. כלומר במקרה שלנו עבור 1000 איטרציות יהיה מעבר בסה"כ על: 2048\*1000=2,048,000.

הקוד שבאמצעותו מימשתי את השיטה נמצא בקובץ PY המצורף.

כאשר A ו B הן רשימות המכילות את ערכי הפרמטרים של השיפוע והחותך. LOSS היא רשימה המכילה את ערכי פונקציית האובדן. ו erorlstהיא רשימה שמייצגת את האיטרציות היא נועדה בשביל ליצור גרף טוב.

**תוצאות הפונקציה עבור קצב למידה=0.0001:**





כפי שניתן לראות בגרף האובדן ישנה התכנסות די מהירה סביב טעות נמוכה(השיפוע הגדול) ואז תיקונים קטנים. זה מחזק את הטענה ממוקדם שבשיטה זאת יש תחילה צעדים גדולים ולאחר מכן צעדים קטנים.

כמו כן עבור השיפוע יש קפיצה מהירה לערך באיזור 5.45 ולאחר מכן תיקונים קטנים עד בערך 5.4.

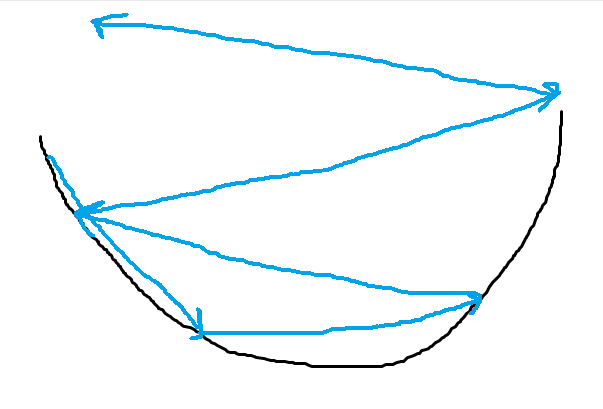
לעומת זאת, החותך מתעדכן באופן תמידי עד שמגיע לערך קצת יותר גבוה מ1.2.

**תוצאות הפונקציה עבור קצב למידה=0.1:**

כשניסיתי להריץ עבור קצב למידה גבוה נזרקה שגיאה:

OverflowError: (34, 'Result too large')

שגיאה זו מצביעה על כך שקצב הלמידה, כלומר הצעדים שנלקחו היו גדולים מידי ולכן הפונקציה אינה מתכנסת לנקודות מינימום אלא שואפת לאינסוף. מכיוון שהצעדים גדולים מידי ישנה "בריחה" מהפרבולה כלפי הכיוון ההפוך מנקודות המינימום.



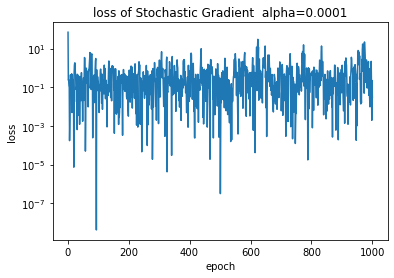
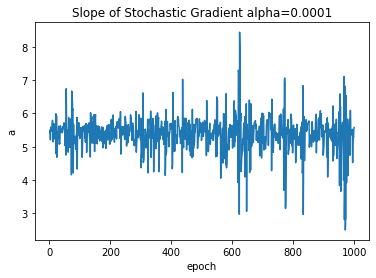
ציור להמחשה של התהליך שקורה, החצים הכחולים מתארים את הצעדים והבריחה מהפונקציה שמצוירת בשחור.

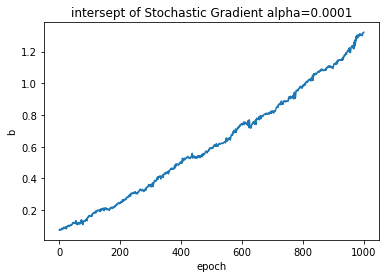
**Stochastic Gradient Descent(SGD)**

שיטה זו היא השיטה המהירה ביותר והכי פחות מדויקת. בשיטה זאת נשתמש כאשר יש לנו DATA מאוד גדולה ונעדיף להוציא תוצאות מהירות יותר אך פחות מדויקות. בשיטה זו נבחרת נקודה רנדומאלית שממנה המודל לומד. כלומר מספר הריצות יהיה כמספר האיטרציות שנגדיר. במקרה של העבודה הפונקציה הזאת תרוץ 1000 פעמים(לעומת 2,048,000 פעמים עבור BGD). הרבה יותר מהיר הרבה פחות מדויק.

הקוד שבאמצעותו מימשתי את השיטה נמצא בקובץ PY המצורף.

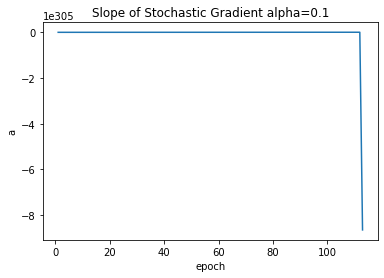
כאשר A ו B הן רשימות המכילות את ערכי הפרמטרים של השיפוע והחותך. LOSS היא רשימה המכילה את ערכי פונקציית האובדן. ו erorlstהיא רשימה שמייצגת את האיטרציות היא נועדה בשביל ליצור גרף טוב.

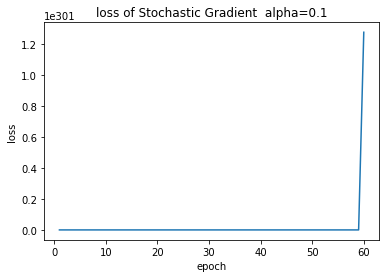
**תוצאות הפונקציה עבור קצב למידה=0.0001:**

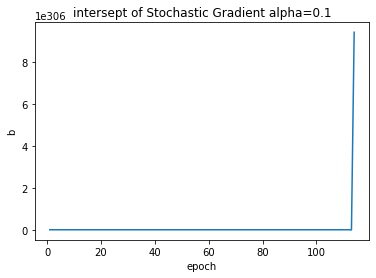


עבור פונקציית האובדן ניתן לראות שיש הרבה רעש,

כלומר הרבה רנדומאליות ניתן גם לראות בגרף שהשגיאות נעות בין בטווח הרבה יותר גדול מאשר בBGD. באופן דומה ניתן לראות שגם השיפוע נע בערכים רנדומאליים בטווח גדול לעומת BGD שבה הוא התכנס ברוב בריצות סביב הטווח שבין 5.4-5.5. במקרה הסטוכסטי הוא נע בין 3-8.גם עבור החוץ ניתן לראות שיש רעש על הקו הליניארי לעומת BGD שבו הקו הליניארי היה ישר. יש לציין שבכל פעם שארית מחדש את המודל הזה אקבל תוצאות אחרות עוד דבר שבמצביע באופן מוחלט על רנדומאליות.

**תוצאות הפונקציה עבור קצב למידה=0.1:**





במקרה זה ניתן לראות שלאחר 60 איטרציות פונקציית העלות מתבדרת, שאופת לאינסוף

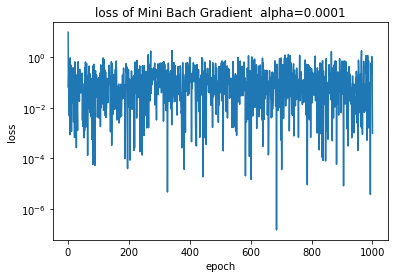
ועבור 120 איטרציות השיפוע שואף למינוס אינסוף והחותך לאינסוף. ניתן להסיק מכך שקצב הלמידה גדול מידי וממש כמו במקרה של BGD גם כאן יש התדברות. אולם, במקרה הסטוכסטי היא מתבדרת לאט יותר מבמקרה של BGD ולכן כאן לא קיבלתי שגיאה. אלא הריצה נעצרה.

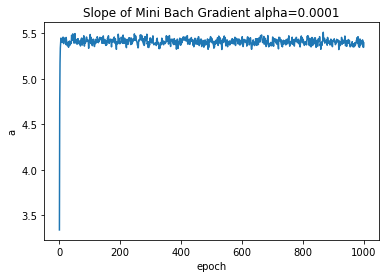
**Mini Bach Gradient Descent (MBGD)**

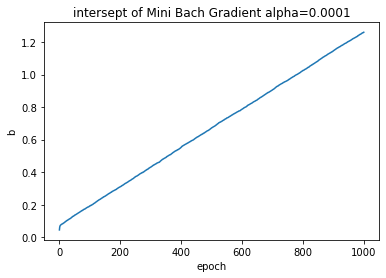
שיטה זאת היא שילוב של שתי השיטות שציינתי למעלה. בשיטה זאת בוחרים באופן אקראי קבוצה של שורות מהDATA ובכך מעדכנים את הפרמטרים. משתמשים בשיטה זאת כדי להוציא תוצאות טובות בזמן יחסית קצר. ככל שנבחר גודל קבוצה גדול יותר נהיה מדויקים יותר אך פחות מהירים. אם נבחר בגודל של 2048 כלומר גודל הDATA במקרה שלנו נקבל תוצאה דומה לBGD ואם נבחר בגודל קבוצה של 1 זה למעשה המקרה הסטוכסטי. שיטה זאת מאוד נפוצה מכיוון שהיא נותנת את האפשרות לתמרן בין מהירות לדיוק. בדוגמה שבחרתי מספר הריצות יהיה: 1000\*64=64000 . ( 2048/32=64)

הקוד שבאמצעותו מימשתי את השיטה נמצא בקובץ PY המצורף.

כאשר A ו B הן רשימות המכילות את ערכי הפרמטרים של השיפוע והחותך. LOSS היא רשימה המכילה את ערכי פונקציית האובדן. erorlst היא רשימה שמייצגת את האיטרציות היא נועדה בשביל ליצור גרף טוב. BatchSize מציין את גודל הקבוצה.

**תוצאות הפונקציה עבור קצב למידה=0.0001 וגודל קבוצה של 32:**





במקרה זה ניתן לראות על פונקציית העלות שישנה רנדומאליות בדומה למקרה הסטוכסטי. גרף השיפוע מתייצב על הערכים שציפתי לקבל ממש בדומה לגרף השיפוע בBGD. אולם אם יותר רעש. הגרף של החותך גם דומה מאוד לגרף של BGD. אולם הוא יותר רועש ממנו.

לסיכום ניתן לראות שפרט לפונקציית האובדן(הרבה יותר רועשת מBGD) בגלל שהיא רנדומאלית. ערכי הפרמטרים של a & b דומים מאוד לערכי הפרמטרים של BGD.

**תוצאות הפונקציה עבור קצב למידה=0.0001 וגודל קבוצה של 32:**

OverflowError: (34, 'Result too large')

גם במקרה הזה כמו בBGD קיבלתי מקרה של התבדרות מאותה הסיבה שנימקתי למעלה.