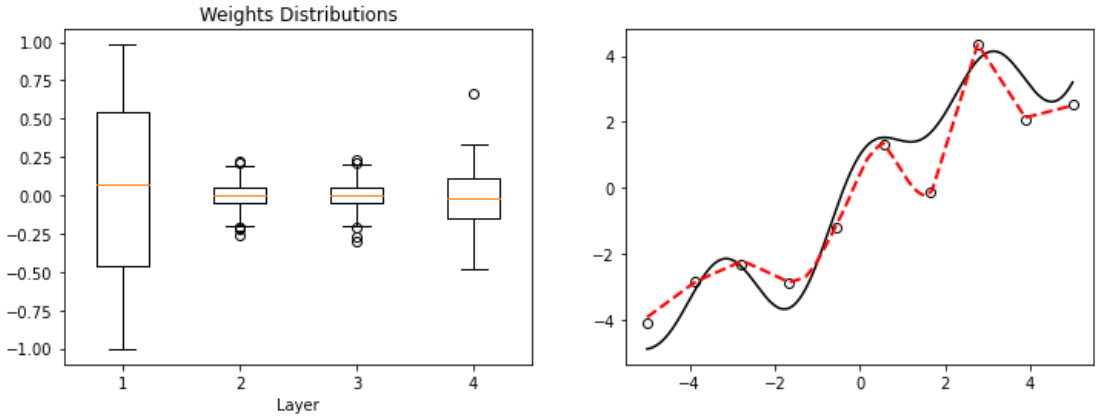
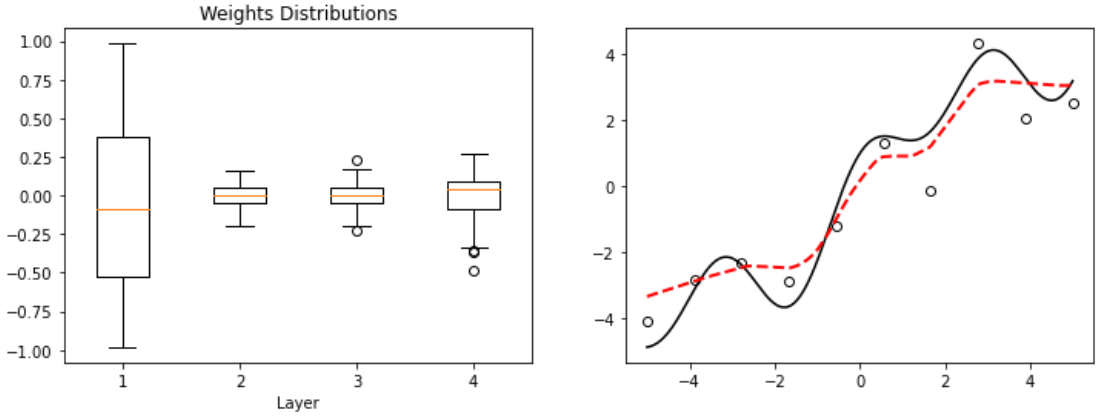
דעיכת המשקלים

באופן קלאסי, הפתרון לבעיית התאמת היתר באלגוריתמי למידת מכונה הוא העדפת מודלים "פשוטים" על פני "מורכבים": כאלו אשר אינם גמישים מספיק בכדי לשנן את סט הלמידה. בהקשר של רשתות נוירונים עקרון זה בא לידי ביטוי על ידי העדפת רשת המפעילה פחות נוירונים, בעוצמה נמוכה יותר מאשר רשת אחרת. ראו בדוגמה של הפרק הקודם, את התפלגויות המשקלים בכל שכבה ברשת אשר שיננה את סט האימון,



לעומת התפלגויות המשקלים של הרשת אשר עצרנו את אימונה מוקדם,



באיורים אלו סיכמנו את התפלגויות המשקלים בצורת Boxplot אשר תכונותיו הן:

- 75% מהדגימות נמצאות בתוך הקופסה המרכזית.
- החציון מסומן בקו באמצע הקופסה.
- לקופסה "שפמים" (whiskers) אשר מגיעים עד לדגימה הגדולה/קטנה ביותר באוסף הנתונים אשר מרוחקת מהקופסה לכל היותר 1.5 פעמים גודל הקופסה.
- דגימות רחוקות יותר מסומנות בנקודות לבנות.

ניתן לצייר Boxplot ולשלוט בפרמטרים השונים בעזרת הפונקציה `boxplot` של הספרייה matplotlib.

מהאיורים ניכר בבירור הבדל בהתפלגות המשקלים בשתי הרשתות: ברשת בעלת התאמת היתר ההתפלגות רחבה יותר: שפמי ה-Boxplot רחבים במעט וכן יש מספר גדול יותר של ערכים חריגים הנמצאים מעבר לשפמים. בפרק זה נלמד שיטות רגולריזציה נוספת אשר כופות על הרשת לכבות/להנמיך את העוצמה של משקלים שאינם תורמים באופן משמעותי להורדת השגיאה של הרשת: נוסיף לפונקציית המחיר של הרשת קנס על מורכבות. אם בעבר פונקציית המחיר אמדה את מידת ההתאמה של הרשת לאוסף הנתונים בלבד, למשל ממוצע הריבועים במקרה של בעיית רגרסיה,

$$C(\text{Model}) = MSE(\text{Model}),$$

כעת לפונקציית המחיר אשר אלגוריתם האופטימיזציה ימזער יתווסף רכיב רגולריזציה, המתגמל מודלים בהם ערכי הפרמטרים קטנים יותר. בהתאם, מודלים אשר מתאימים במיוחד לסט האימון אך

בעלי ערכי פרמטרים גדולים יידחו בתהליך האופטימיזציה. כך למעשה אנו מצמצמים את תחום החיפוש למודלים אשר ערכי הפרמטרים שלהם קרובים לראשית בלבד.

$$C(\text{Model}) = MSE(\text{Model}) + \lambda \cdot Size(\text{Weights}).$$

הפרמטר λ (שהינו תמיד אי שלילי) מאזן בין שתי המטרות השונות של פונקציית המחיר החדשה – התאמה לנתונים והעדפת מודל פשוט. עבור $\lambda = 0$ לא תתבצע רגולריזציה כלל, ועבור λ גדול איפוס הפרמטרים ישתלט על תהליך האופטימיזציה, והרשת לא תלמד כלל.

רגולריזציית L_2

אחת האפשרויות הפופולריות למדד הגודל של הפרמטרים הוא נורמת L_2 של המשקלים (למעשה אנו מעדיפים את מחצית ריבוע הנורמה, שכן כך החישובים בהמשך פשוטים יותר):

$$\frac{1}{2} L_2(\text{Weights})^2 = \frac{1}{2} \sum_{w \in \text{Weights}} w^2.$$

בשיטת רגולריזציה זו יתווסף בכל צעד באלגוריתם האופטימיזציה רכיב הרגולריזציה לחישוב הגרדיאנט:

$$\frac{\partial C}{\partial w} = \frac{\partial MSE}{\partial w} + \frac{\partial}{\partial w}\left(\frac{\lambda}{2} L_2\right) = \frac{\partial MSE}{\partial w} + \lambda w.$$

על כן הפרמטר w יעודכן על ידי SGD באופן הבא:

$$w_{t+1} = w_t - \alpha\left(\frac{\partial MSE}{\partial w} + \lambda w_t\right) = (1 - \alpha\lambda) w_t - \alpha \frac{\partial MSE}{\partial w}$$

מנוסחה זו ניתן להבין את השם שהשיטה קיבלה בקהילת הלמידה העמוקה – דעיכת המשקלים (Weight decay): בכל איטרציה המשקל w מאבד אחוז מסויים מערכו, ודועך אל האפס. יש לשים לב כעת ש- λ משפיע על גודל צעד העדכון של אלגוריתם SGD: עבור ערכי λ גדולים מדי ייתכן שהאלגוריתם כבר לא יתכנס, עקב קפיצות גדולות מדי בין האיטרציות, תופעה הדומה לבחירת גודל צעד α גדול מדי.

נממש שיטה זו בקוד כעת. המתודה `named_parameters()` של המודל מחזירה איטרטור על כל הפרמטרים הקיימים ברשת יחד עם שמותיהם, ובמעבר על כולם, נחלץ רק את **המשקלים** לרשימה אחת. אלו הפרמטרים אשר עליהם נפעיל רגולריזציה, שכן לרוב אין אנו מעוניינים לבצע רגולריזציה על ערכי ה-bias של הנוירונים – הסוג השני של פרמטרים ברשת הנוכחית.

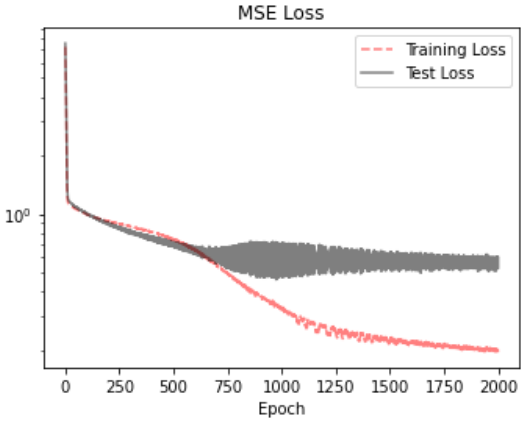
```
weights = [param for (name,param) in model.named_parameters()
           if "weight" in name]
```

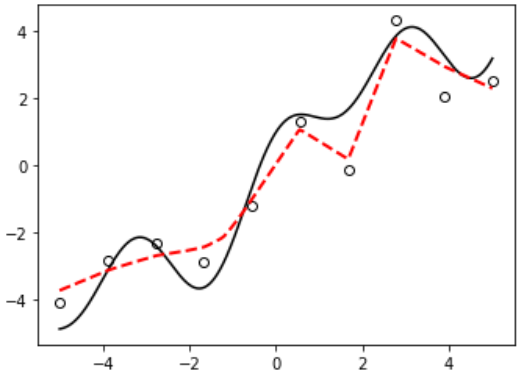
אחרי כן, ללולאת האימון נוסיף את חישוב רכיב הרגולריזציה, ולבסוף נחשב את הגרדיאנט של פונקציית המחיר החדשה, המורכבת משני החלקים.

```
reg_loss=0
  for param in weights:
    reg_loss += (param**2).sum()
  total_loss = MSE_loss + 1/2*lambd*reg_loss
  total_loss.backward()
```

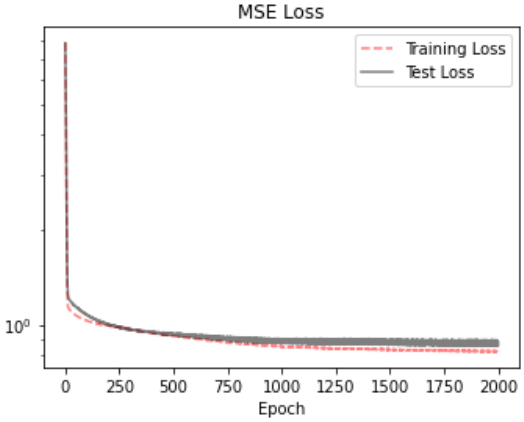
המשך תהליך אימון הרשת נשאר כשהיה.

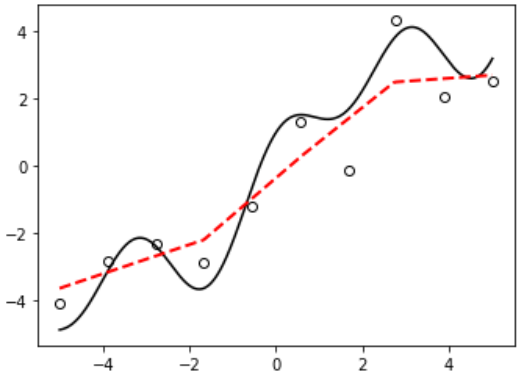
תוצאות תהליך האימון, עבור ערך λ קטן מאויירת להלן, ובהן ניתן לראות שעדיין מתבצעת התאמת יתר לאורך דורות האימון, אך השפעתה פחותה מבעבר.



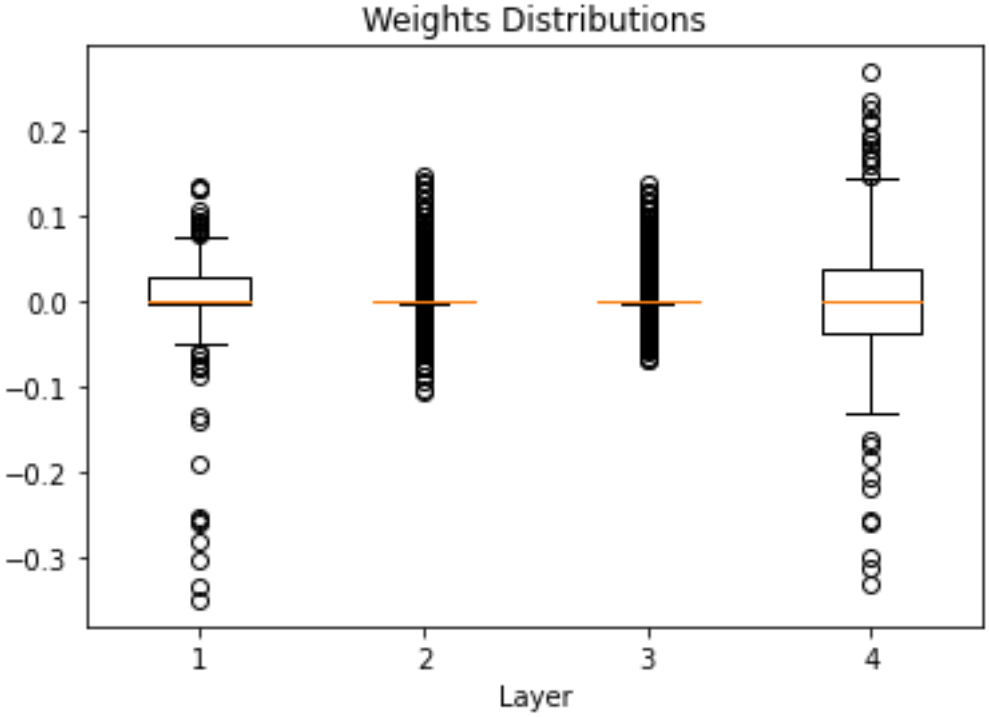


עבור ערך λ גדול יותר, תתקבל התוצאה הבאה, וממנה ניכר שאין התאמת יתר משמעותית, במחיר פשטות יתר של המודל המתקבל.





במבט מעמיק יותר לתוצאת הריצה האחרונה, ניכר שרוב המשקלים בשכבות הפנימיות ברשת כמעט התאפסו, שכן תרומתם להורדת מחיר ההתאמה של הרשת לנתונים לא היתה מספיק משמעותית בהשוואה לגורם הדעיכה של הרגולריזציה. ראו זאת באיור ההתפלגויות להלן.



רגולריזציית L_1

שיטה נוספת פופולרית ביותר לרגולריזציה היא שימוש בנורמת L_1 של המשקלים, כך שפונקציית המחיר החדשה אותה יש למזער תהיה

$$C(\text{Model}) = MSE(\text{Model}) + \lambda \sum_{w \in \text{Weights}} |w| .$$

בדומה לשיטה הקודמת, נתבונן בהשפעת השיטה על גרדיאנט פונקציית המחיר:

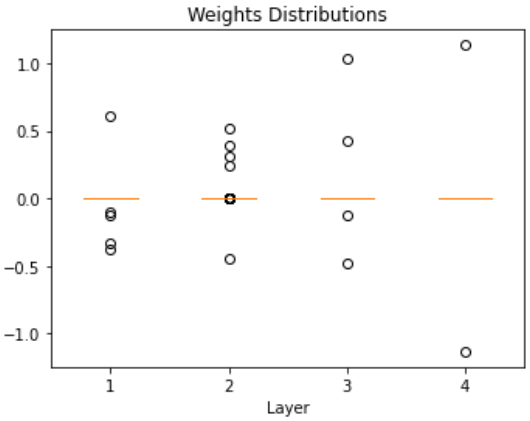
$$\frac{\partial C}{\partial w} = \frac{\partial MSE}{\partial w} + \lambda \operatorname{sgn}(w)$$

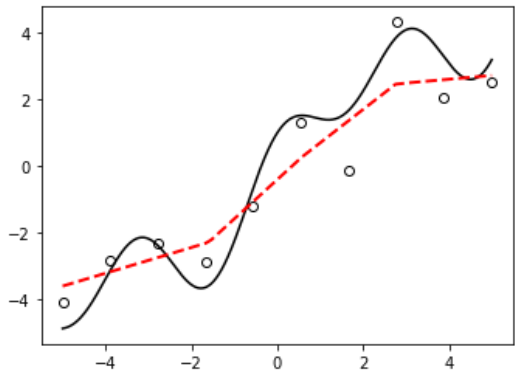
כאשר $\operatorname{sgn}(w)$ היא פונקציית הסימן. שימו לב ששוב התעלמנו מהבעיה בחישוב הנגזרת של $|w|$ בנקודה 0, בדומה לאקטיבציה $\operatorname{ReLU}(x)$. צעד העדכון ב-SGD יהיה:

$$w_{t+1} = w_t - \alpha\left(\frac{\partial MSE}{\partial w} + \lambda \operatorname{sgn}(w)\right) = w_t - \alpha \frac{\partial MSE}{\partial w} - \alpha\lambda \operatorname{sgn}(w)$$

למרות הדימיון הרב בין שתי השיטות, יש לשים לב לכך שברגולריזציית L_1, דעיכת הפרמטר w לכיוון האפס אינה פרופורציונית לגודלו: בכל איטרציה מתבצע צעד קבוע בגודל $\alpha\lambda$. כמו במקרה הקודם, יש לשים לב לבחירה שקולה של גודל הצעד α ופרמטר הרגולריזציה λ כך שמכפלתם לא תהיה גדולה מדי, אחרת הקפיצות הקבועות בגודל זה לכיוון האפס יובילו לתנועה מחזורית ואלגוריתם האופטימיזציה לא יתכנס.

גודל הצעד הקבוע משפיע באופן דרמטי על פרמטרים קטנים בגודלם, איך השפעתו זניחה על פרמטרים גדולים. התוצאה המתקבלת היא שהרשת תרכז את רוב המשקל על מספר מועט של משקלים, והשאר יתאפסו לחלוטין. ראו דוגמה לכך בתוצאת אימון רשת הרגרסיה יחד עם שיטת רגולריזציה זו באיור הבא.





בבואנו לממש שיטה זו בקוד, כל שיש לעשות הוא לחשב את מחיר הרגולריזציה בדרך שונה:

```
reg_loss=0
  for param in weights:
    reg_loss += param.abs().sum()
  total_loss = MSE_loss + lambd*reg_loss
```

את ההבדל בין התוצאות המתקבלות משתי גישות אלו לרגולריזציה ניתן להבין גם דרך דוגמה פשוטה מאוד, בה פרמטרי המודל הם $w_0,\ldots,w_n$ ופונקציית המחיר בעלת הצורה הבאה:

$$Loss\left(w_0,...,w_n\right)=\frac{1}{2}\sum_{k=0}^{n}H_k\left(w_k-w_k^*\right)^2$$

כאשר $H_k>0$ לכל k. אומנם מקרה זה רחוק מאוד מפונקציות המחיר המורכבות של רשתות נוירונים עמוקות, אך הוא אינו מנותק מהמציאות: פונקציה כזו או דומה לה מופיעות בבעיות רגרסיה ליניארית.

נקודת המינימום (היחידה) של מחיר זה היא כמובן $\left(w_0^*,...,w_n^*\right)$ ובה ערך הפונקציה הוא אפס – אלו פרמטרי המודל האופטימליים. אנו מצפים כי הרגולריזציה תזיז את הערכים האופטימליים אל הראשית, אך ההבדל בין שתי הגישות הוא הדרך בה תנועה זו מתרחשת כאשר הפרמטר λ גדל.
ראשית נתבונן במקרה בו לפונקציית המחיר נוסף גורם של רגולריזציית L_2:

$$C_2\left(w_0,...,w_n\right)=Loss+\frac{\lambda}{2}\sum_{k=0}^{n}w_k^2 .$$

ניתן להוכיח (אך לא נעשה זאת כאן) כי נק' המינימום החדשה, אותה נסמן $\left(\hat{w}_0,...,\hat{w}_n\right)$ מתקבלת מהמקורית לפי הנוסחה הבאה:

$$\hat{w}_k=\frac{H_k}{H_k+\lambda}w_k^* .$$

מכאן נסיק שכל משקל אכן כווץ לכיוון האפס, **באופן כפלי**, אך כאלו בעלי מקדם H_k גדול ביחס ל- λ (אלו המשקלים בעלי השפעה רבה על פונקציית המחיר) כמעט שלא כווצו.

לעומת זאת, עבור רגולריזציית L_1 פונקציית המחיר החדשה היא

$$C_1\left(w_0,...,w_n\right)=Loss+\lambda\sum_{k=0}^{n}\left|w_k\right| ,$$

ונק' המינימום החדשה מתקבלת לפי הנוסחה

$$\hat{w}_k=\operatorname{sgn}\left(w_k^*\right)\cdot\max\left(\left|w_k^*\right|-\frac{\lambda}{H_k},0\right) .$$

נוסחה זו יש להבין כך: אם מרחקו של הפרמטר w_k^* מהאפס היה גדול מ- $\frac{\lambda}{H_k}$, הוא הוזז לכיוון האפס על ידי החסרת/תוספת גורם זה. אם ערכו המקורי היה קטן מדי – הוא פשוט אופס. כאן ניתן לראות בבירור כיצד גישת רגולריזציה זו גורמת לאיפוס פרמטרים, וכיצד גודל הפרמטר המקורי אינו משפיע על עוצמת השינוי בערכו.

שאלות לתרגול

1. הפעילו רגולריזציה על כל פרמטרי רשת הרגרסיה הנ"ל, כולל ערכי ה-bias ואיירו את התוצאות המתקבלות. האם דרך דוגמה זו תוכלו להסביר למה נהוג לא להפעיל עליהם רגולריזציה?
2. אמנו את רשת סיווג תמונות פרטי הלבוש בסט הנתונים Fashion-MNIST עם רגולריזציית L_1. מצאו את ערך הפרמטר λ האידיאלי המאזן בין פשטות המודל לבין התאמת יתר לסט האימון. **רמז:** חפשו את הפרמטר λ עבורו מתקבלת שגיאת ההכללה הנמוכה ביותר.
3. הסבירו למה בשאלה הקודמת יש צורך בשימוש בסט נתוני ולידציה וסט נתוני בדיקה להערכת ביצועי הרשת באופן אמין.