תרגיל פייתון – מבוא ללמידה מכונה סטטיסטית

מגיש: Matan Adar מתן אדר

ת.ז: 322357542

חלק 1 וחלק 3 השוואה- בהמשך הקובץ

חלק 1:

אציג גרפי שגיאה כתלות במס' ה- epochs על פני סט האימון וסט הבדיקה:

- גרפים (מנורמלים) שמתארים את השינויים בין initializations שונים של המשקולות וקצב הלמידה וגרף שמתאר את השינוי בשגיאה היחסית (צירים שונים עבור כל גרף) כתלות במס' ה- epochs בין סט האימון וסט הבדיקה.

- גרף אחוד, צבוע לפי קצבי למידה

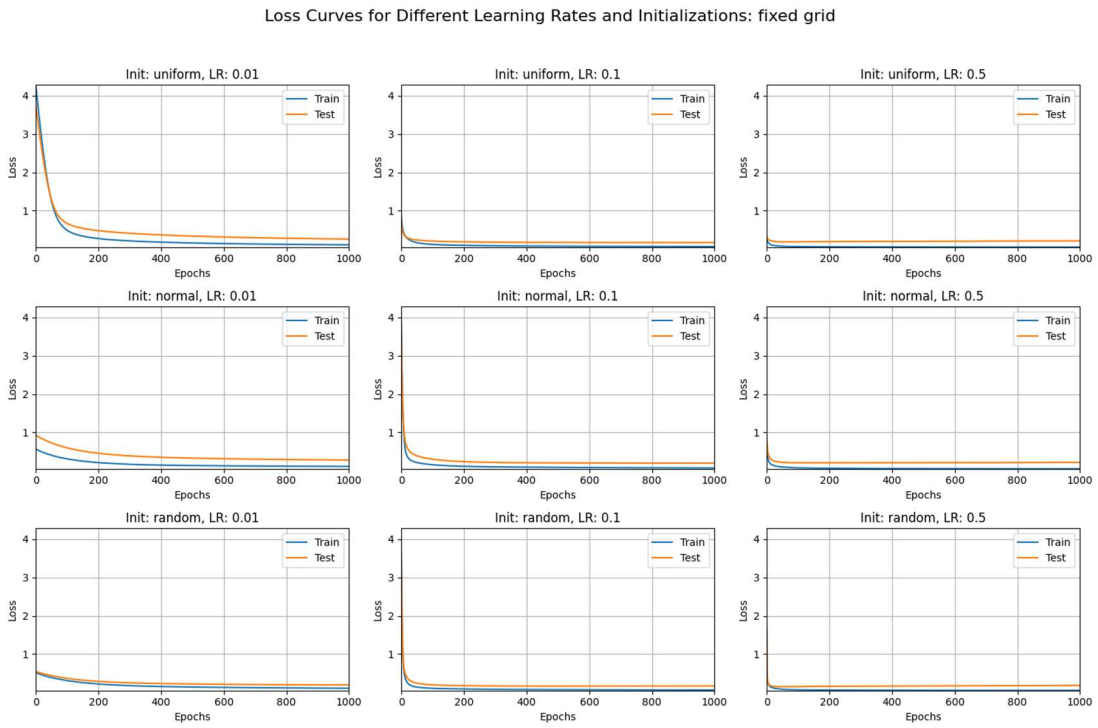
- גרף אחוד, צבוע לפי סוגי אתחול

להלן טבלה שמייצגת את התוצאות:

Learning Rates		uniform		normal		random	
		train	test	train	test	train	test
0.01	Final loss	0.1071	0.2551	0.1104	0.2752	0.1065	0.1936
	Final accuracy	0.9538	0.8772	0.9582	0.8684	0.9648	0.9386
0.1	Final loss	0.0614	0.1660	0.0689	0.1948	0.0609	0.1663
	Final accuracy	0.9802	0.9386	0.9736	0.9035	0.9802	0.9298
0.5	Final loss	0.0468	0.2091	0.0475	0.2150	0.0495	0.1806
	Final accuracy	0.9846	0.9035	0.9846	0.9035	0.9846	0.9298

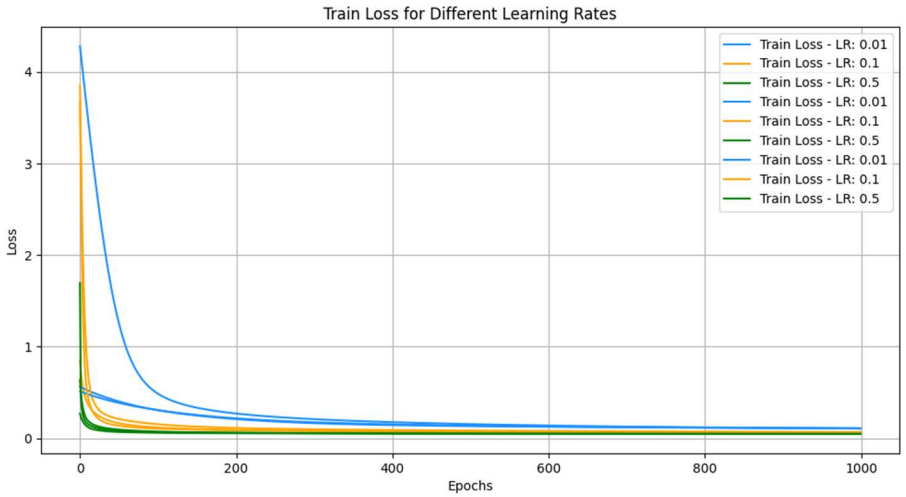
אפשר לראות שהמודל עם הביצועים הטובים ביותר הוא המודל בו המשקולות אותחלו לפי התפלגות אחידה, וקצב הלמידה שלו 0.1. אין בהכרח התאמה בין ערך הדיוק לערך השגיאה, המודל הכי גרוע הוא לא בהכרח המודל שעבורו ההפסד הכי גבוה.

גרפים מנורמלים:

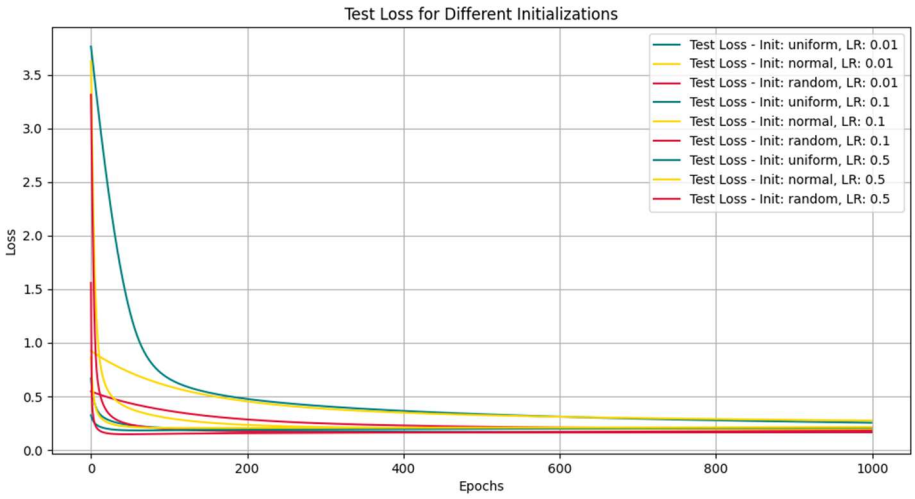


ניתן לראות שככל שקצב הלמידה עולה, מספר ה- epochs השגיאה מתכנסת לערך שהולך וקטן- תהליך הלמידה מהיר יותר. אז במקום תנאי העצירה שבחרנו (1000 epochs) יכולנו לבחור תנאי עצירה מוקדם יותר עבור שגיאות שקטנות מערך ספציפי וכך סיבוכיות זמן הריצה של הקוד תקטן (פחות איטרציות).

ניתן לראות זאת בגרף הבא שבו הפונקציות נצבעו לפי קצב הלמידה-



בגרף שבו הפונקציות נצבעו לפי סוג האתחול לא כל כך פשוט לקבוע מה ההשפעה של סוג האתחול על התנהגות הפונקציה-



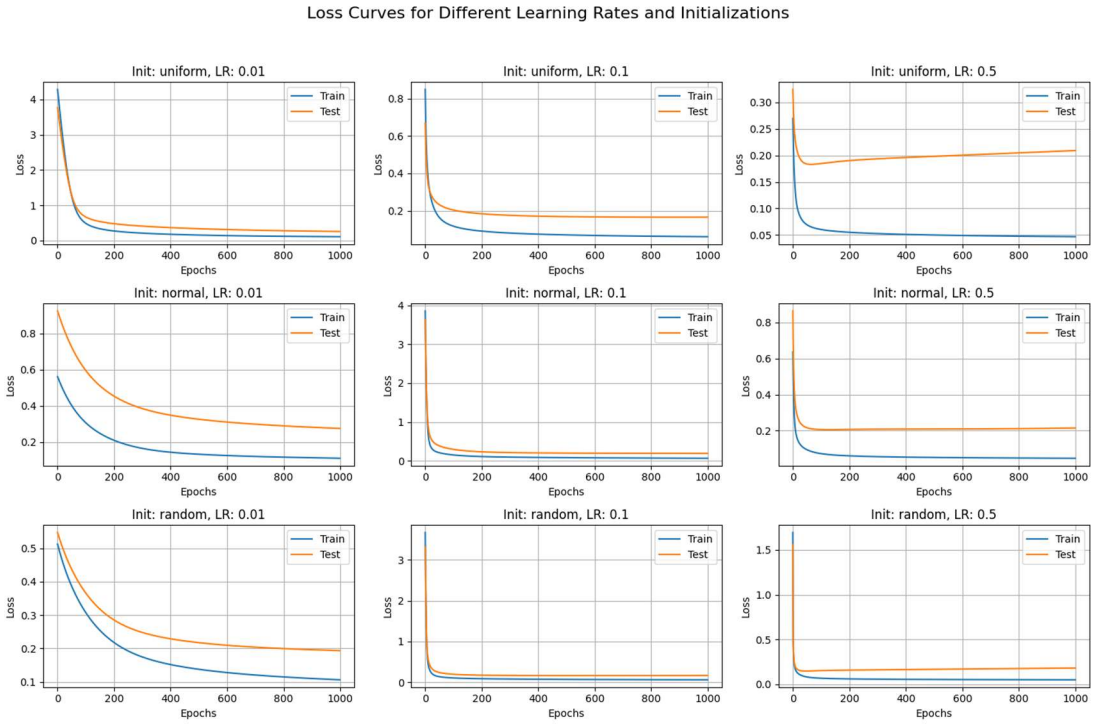
נשים לב שלכל סוג אתחול יש מספר נמוך יחסית של דגימות. אם היו יותר בדיקות של יותר ערכים עבור קצב הלמידה היינו יכולים להבין את ההשפעה של האתחול.

Model with the Lowest Final Test Loss (and Highest Test Accuracy=0.1)

Init Type: uniform
Learning Rate: 0.1

Metric	Value
Final Train Loss	0.0614
Final Test Loss	0.1660
Biases	[[0.03760685]]
Weights	[[-0.32144205 -1.33961924 -0.70967423 -0.29696062 -0.51612673 0.83709896 -1.09992213 -0.65535655 -0.38355072 0.04896723 -0.59537995 0.08458585 0.09824199 -0.99252315 -0.13918601 0.26549082 1.10200913 -0.25619367 0.45709878 -0.11344414 -1.56299044 -0.62685338 -0.44921112 -0.85601717 -0.51548253 0.20072985 -0.78049687 -1.45276549 -0.51721723 -0.0346638]]

טבלאות של גרפים לא מנורמלים-



מצרף נוסחאות רלוונטיות לקוד-

(◇) Useful property of the sigmoid:

$$\sigma'(z) = \sigma(z)(1 - \sigma(z))$$

Cross-entropy loss:

$$L = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

- m: Number of data samples in the training set
- y_i: The true label of the i-th sample
- $\hat{y}_i = \sigma(w^T x_i + b)$: The predicted output for the i-th sample

---

$$\frac{\partial L}{\partial b} = -\frac{1}{m}\sum_{i=1}^{m}\frac{\partial}{\partial b}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\frac{\partial \hat{y}_i}{\partial b}\left[\frac{y_i}{\hat{y}_i} - \frac{1 - y_i}{1 - \hat{y}_i}\right]$$

$$\frac{\partial \hat{y}_i}{\partial b} = \sigma'(w^T x_i + b) = \hat{y}_i(1 - \hat{y}_i)$$

$$\Rightarrow \frac{\partial L}{\partial b} = -\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)$$

$$\frac{\partial L}{\partial w} = -\frac{1}{m}\sum_{i=1}^{m}\frac{\partial}{\partial w}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\frac{\partial \hat{y}_i}{\partial w}\left[\frac{y_i}{\hat{y}_i} - \frac{1 - y_i}{1 - \hat{y}_i}\right]$$

$$\frac{\partial \hat{y}_i}{\partial w} = \hat{y}_i(1-\hat{y}_i)x_i$$

$$\Rightarrow \frac{\partial L}{\partial w} = -\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)x_i$$

חלק 3

השוואה-

כמתבקש פיצלנו את הדאטה סט ל- 80% אחוז אימון ו-20% טסטים עבור כל אחד מהמודלים.

אימנו את האלגוריתמים על הדאטה, וההשוואה בין המודלים מתבסס על הדאטה.

נבחן מספר פרמטרים-

אחוזי דיוק-

עבור העץ התקבל אחוז דיוק של-

Accuracy: 92.98%

(הדיוק משתנה כתלות בצורת בחירת ערכים רנדומליים בקוד)

עבור רשת הנוירונים, אחוז הדיוק תלוי בקצב הלמידה ובאתחול המשתנים, אך ראינו שעבור התפלגות אחידה וקצב למידה של 0.1 התקבל אחוז דיוק של 93.86% שטוב מאחוז הדיוק של העץ **(כדאי לשים לב, אחרי שמריצים שוב ושוב את התוכנית לרוב מתקבל אחוז דיוק גבוה יותר ברשת הנוירונים).**

התבוננות בשיטת התיוג, הסבר ודרך ההצגה של המודלים-

בחלק 2- בדקתי עבור כל פרמטר מה הערך שלו, לאיזה כיוון בעץ הוא צריך לרדת, עד שהגענו לסוף העץ, לסוף הנתיב וקיבלנו החלטה. די פשוט להסביר למה דגימה מסוימת מתויגת כך או אחרת. בניגוד לכך, ברשת הנוירונים, הבנת התיוג של דגימה מסוימת לא טבעית, לא טריוויאלית. את העץ קל להציג באופן טבעי, לעומת רשת הנוירונים שאין דרך להציג ויזואלית.

סיבוכיות-

מההרצאה, לרוב רשתות נוירונים הן יותר כבדות חישובית לעומת עץ, והדבר מהווה בעיה לא קטנה עבור דאטה גדולה ומסובכת. עבור הדאטה הנתונה, ההבדלים בין המודלים לא היו גדולים בתחום הזה.

לסיכום-

רשת נוירונים מציעה יתרונות גדולים בזיהוי תבניות מורכבות, ולכן היא מצליחה להתמודד עם בעיות לא לינאריות ולהפיק ביצועים טובים יותר כאשר מדובר בנתונים גדולים ומורכבים. מצד שני, החיסרון העיקרי של רשת נוירונים הוא שדרישות החישוב שלה גבוהות ודורשות משאבים רבים, כמו גם זמן אימון ארוך.

עץ החלטות ,לעומת זאת, מציע יתרון בנוגע לפשטות והבנת המודל: כל החלטה בעץ ברורה ומובנת, ויש אפשרות בקלות לעקוב אחרי כל שלב בתהליך קבלת ההחלטות. עץ החלטות הוא גם מהיר יותר מבחינת חישוב, אך הוא עלול להיתקל בבעיות של אוברפיטינג כאשר יש נתונים מורכבים מאוד או בעיות לא לינאריות, ומכאן ייתכן שיבצע פחות טוב על נתונים חדשים או בעיות מורכבות.

במקרה שלי, לדעתי רשת נוירונים היא עדיפה מכיוון שהיא מצליחה להפיק אחוזי דיוק גבוהים יותר מהשיטה של עץ החלטות, כאשר היא מסוגלת להתמודד עם הדאטה-סט שנתון לה בצורה יעילה.

מצד שני, אם נעדיף דרך הצגה פשוטה של המודל, או להפחית בכובד חישובי, נוכל לבחור את העץ.