למידת מכונה תרגיל 4 – אורי דאבוש

חלק א':

בחלק זה מימשתי את המודלים השונים שהתבקשתי לממש בתרגיל. לכל מודל אתאר כמה דברים:

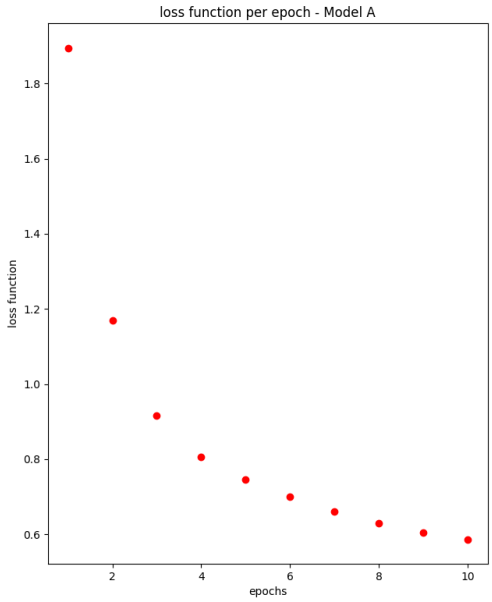
1. גרף שמתאר את ה-loss הממוצע בכל epoch
2. גרף שמתאר את ה-accuracy בכל epoch
3. ה-accuracy של המודל אחרי האימון על ה-FashionMNIST dataset שמספקת לנו החבילה PyTorch.
4. היפר פרמטרים שונים שהשתמשתי בהם במודל.

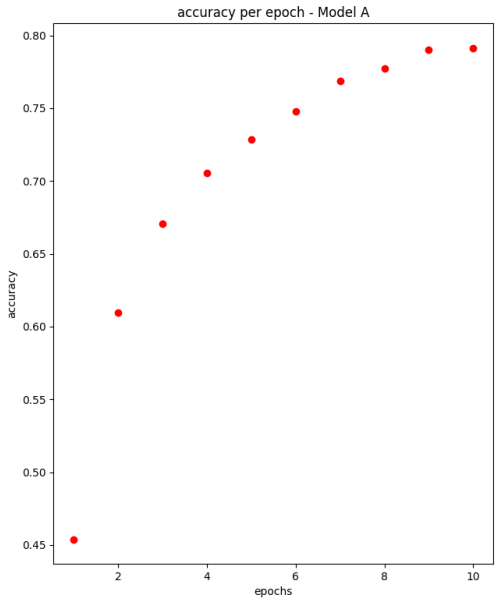
כל המודלים מתאמנים על ה-44,000 (80%) שורות הראשונות של הקובץ train_x, ואני בוחן את סעיפים 1 ו-2 על 11,000 השורות הנותרות (20%).

בנוסף, כל המודלים מתאמנים למשך 10 איפוקים (כפי שנדרש בתרגיל), והם משתמשים בנתונים מנורמלים על ידי חלוקה ב-255 (כלומר נרמול לטווח [0,1]).

<u>מודל A:</u>

במודל זה יש שתי שכבות נסתרות – השכבה הראשונה בגודל 100 והשנייה בגודל 50. לאחר כל שכבה פונקציית האקטיבציה היא ReLU. בשכבה האחרונה פונקציית האקטיבציה היא log softmax.
בנוסף, המודל משתמש באופטימייזר SGD.



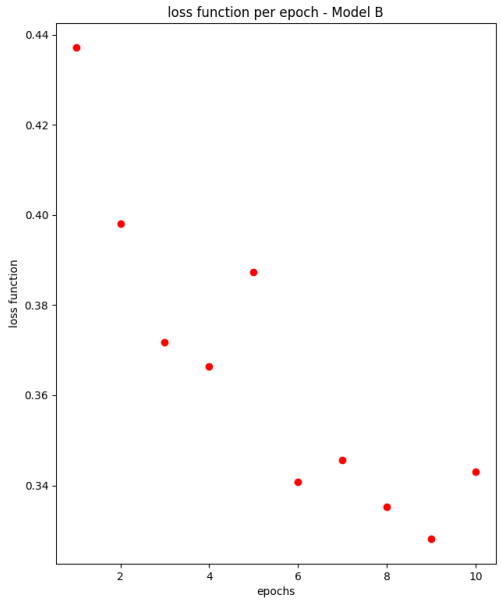


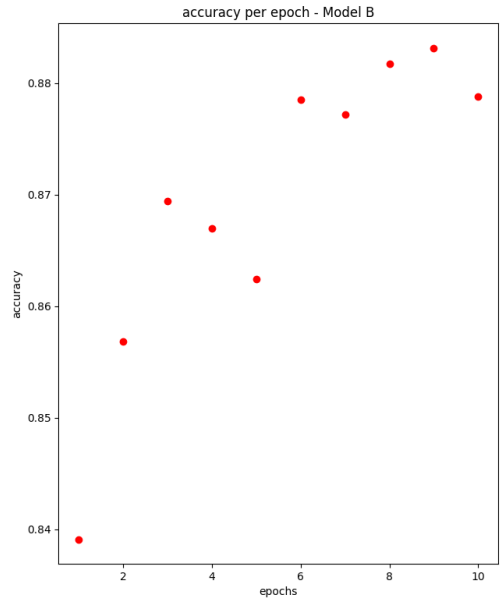
ה-accuracy של המודל הזה על ה- FashionMNIST datase הוא 0.786 (78.6%).

במודל זה ה-learning rate הוא 0.01, וזהו ההיפר פרמטר היחיד שנעשה בו שימוש (פרט לאופטימייזר שהוא קבוע SGD לפי דרישות התרגיל).

<u>מודל B:</u>

במודל זה יש שתי שכבות נסתרות – השכבה הראשונה בגודל 100 והשנייה בגודל 50. לאחר כל שכבה פונקציית האקטיבציה היא ReLU. בשכבה האחרונה פונקציית האקטיבציה היא log softmax.
בנוסף, המודל משתמש באופטימייזר Adam.





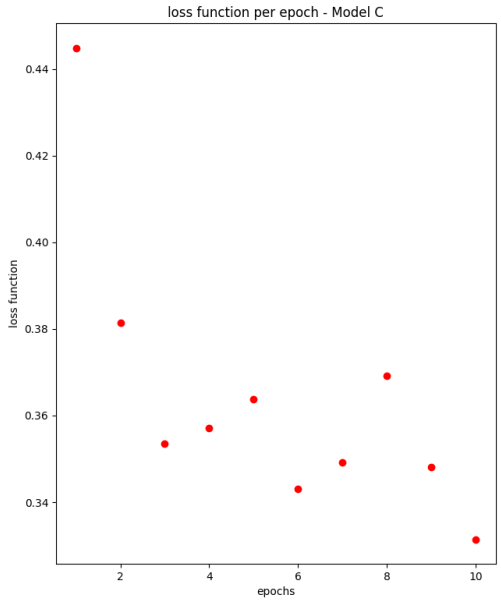
ה-accuracy של המודל הזה על ה- FashionMNIST datase הוא 0.8657 (86.57%).

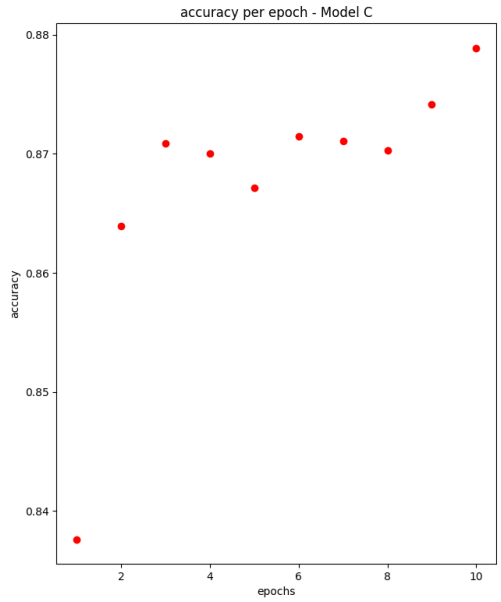
במודל זה ה-learning rate הוא 0.01, וזהו ההיפר פרמטר היחיד שנעשה בו שימוש (פרט לאופטימייזר שהוא קבוע Adam לפי דרישות התרגיל).

<u>מודל C:</u>

במודל זה יש שתי שכבות נסתרות – השכבה הראשונה בגודל 100 והשנייה בגודל 50. לאחר כל שכבה פונקציית האקטיבציה היא ReLU. בשכבה האחרונה פונקציית האקטיבציה היא log softmax. בנוסף, המודל משתמש באופטימייזר Adam.

זה בעצם מודל B עם תוספת אחת – לאחר השכבות הנסתרות מתבצע dropout, כלומר כל תא בוקטור התוצאה של השכבה מתאפס בהסתברות p (ההסתברות p מתוארת בהמשך, בחלק של ההיפר פרמטרים).





ה-accuracy של המודל הזה על ה- FashionMNIST datase הוא 0.8756 (87.56%).

במודל זה ה-learning rate הוא 0.01, וההסתברות p של ה-dropout לאחר שתי השכבות הנסתרות היא 0.05, כלומר בהסתברות 0.05 תא מתאפס בוקטור התוצאה של השכבות הנסתרות.

<u>מודל D:</u>

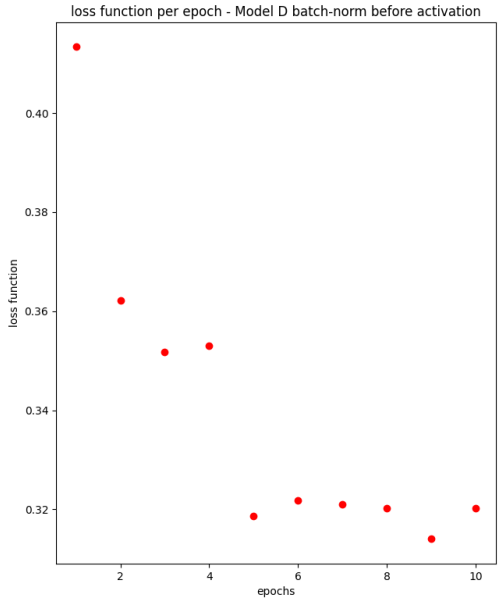
במודל זה יש שתי שכבות נסתרות – השכבה הראשונה בגודל 100 והשנייה בגודל 50. לאחר כל שכבה פונקציית האקטיבציה היא ReLU. בשכבה האחרונה פונקציית האקטיבציה היא log softmax. בנוסף, המודל משתמש באופטימייזר Adam.

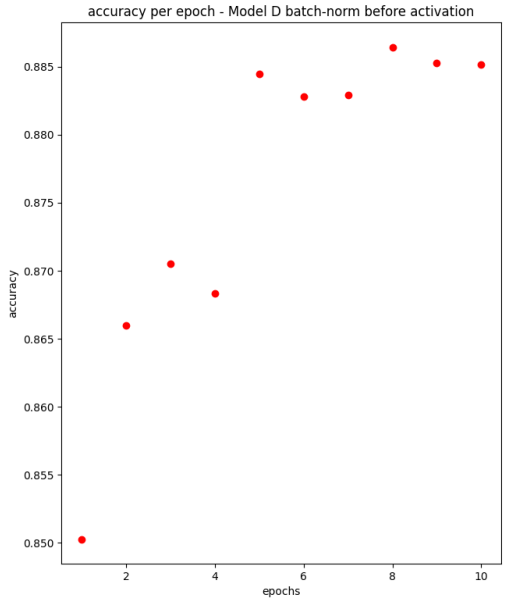
זה בעצם מודל B עם תוספת אחת – מתבצע בו batch normalization, כלומר מנרמלים את הוקטור לפי ממוצע וסטיית תקן. יש לנו שתי אפשרויות – או שנבצע את הנרמול לפני הקריאה לפונקציית האקטיבציה בכל שכבה נסתרת, או שנבצע אחרי.

כדאי לבצע את הנרמול לפני פונקציית האקטיבציה כיוון שפונקציית האקטיבציה ReLU מאפסת חלק מהערכים וייתכן שהנרמול יגרום לאיפוס הערכים האלו, מה שלא היה קורה אם לא היינו מנרמלים לפני הקריאה ל-ReLU.

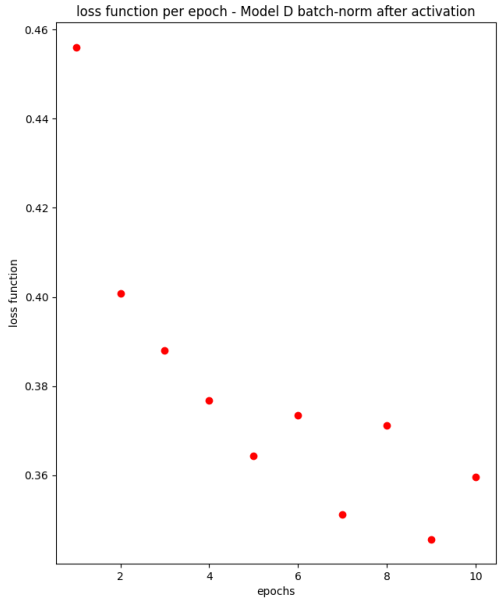
בדקתי את שתי האפשרויות. התוצאות מוצגות בגרפים הבאים.

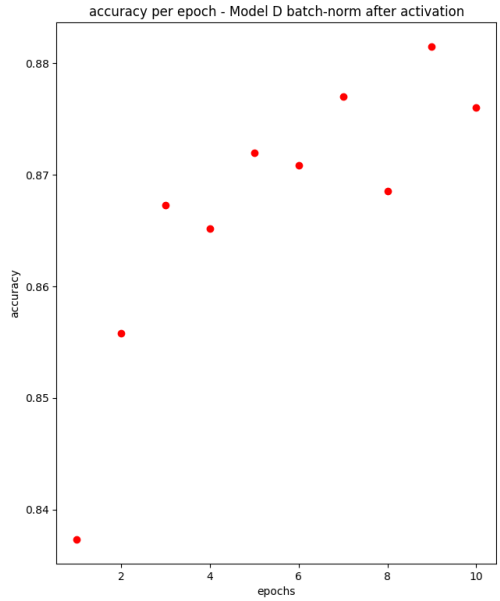
נרמול לפני פונקציית אקטיבציה:





נרמול אחרי פונקציית אקטיבציה:





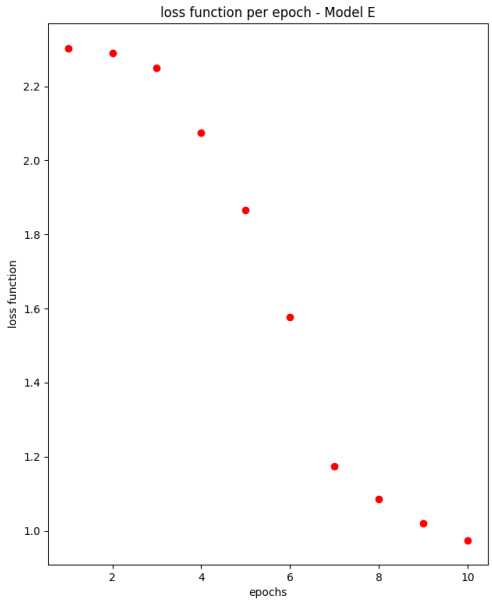
אכן ניתן לראות שהתוצאות של נרמול לפני קריאה לפונקציית אקטיבציה אכן יותר יציבות מהתוצאות של נרמול אחרי קריאה לפונקציית אקטיבציה, למרות שהתוצאות הסופיות במקרה הנ"ל די קרובות.

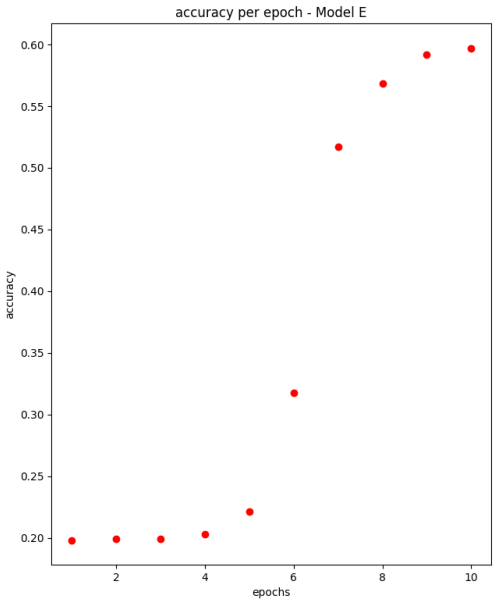
ה-accuracy של המודל הזה על ה- FashionMNIST datase עם נרמול לפני קריאה לפונקציית אקטיבציה הוא 0.8761 (87.61%), ועם נרמול אחרי קריאה לפונקציית אקטיבציה הוא 0.8652 (86.52%).

במודל זה ה-learning rate הוא 0.01, וזהו ההיפר פרמטר היחיד שנעשה בו שימוש (פרט לאופטימייזר שהוא קבוע Adam לפי דרישות התרגיל).

<u>מודל E:</u>

במודל זה יש חמש שכבות נסתרות – והן בגדלים 128, 64, 10, 10, 10 (מימין לשמאל). לאחר כל שכבה פונקציית האקטיבציה היא ReLU. בשכבה האחרונה פונקציית האקטיבציה היא log softmax. בנוסף, המודל משתמש באופטימייזר SGD.



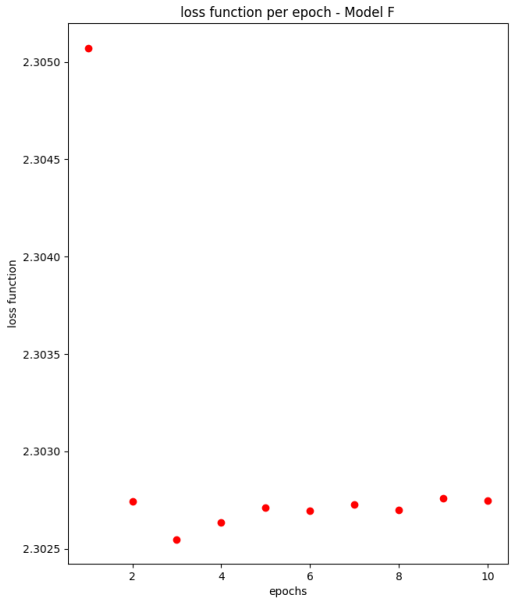


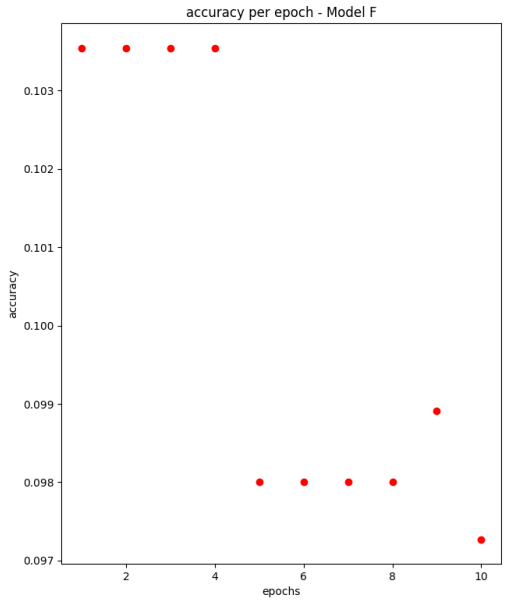
ה-accuracy של המודל הזה על ה- FashionMNIST datase הוא 0.6075 (60.75%).

במודל זה ה-learning rate הוא 0.01, וזהו ההיפר פרמטר היחיד שנעשה בו שימוש (פרט לאופטימייזר שהוא SGD כפי שכבר ציינתי).

<u>מודל E:</u>

במודל זה יש חמש שכבות נסתרות – והן בגדלים 128, 64, 10, 10, 10 (מימין לשמאל). לאחר כל שכבה פונקציית האקטיבציה היא sigmoid. בשכבה האחרונה פונקציית האקטיבציה היא log softmax. בנוסף, המודל משתמש באופטימייזר SGD.





ה-accuracy של המודל הזה על ה- FashionMNIST datase הוא 0.1 (10%).

במודל זה ה-learning rate הוא 0.01, וזהו ההיפר פרמטר היחיד שנעשה בו שימוש (פרט לאופטימייזר שהוא SGD כפי שכבר ציינתי).

חלק ב':

בחלק זה מימשתי מודל משלי (בעזרת החבילה PyTorch כמובן).

המודל גם הוא ממומש באותו אופן כמו המודלים הקודמים, הוא בעל שתי שכבות נסתרות – שכבה ראשונה בגודל 256 והשכבה השנייה בגודל 256.

המודל משתמש באופטימייזר Adam עם learning rate של 0.01.

המודל משתמש בשתי שכבות dropout עם הסתברות 0.2 בשכבה הראשונה ו-0.5 בשכבה השנייה להתאפסות רכיב בוקטור. שכבות ה-dropout הן לאחר השכבות הנסתרות. בנוסף נעשה batch normalization על הנתונים לפני פונקציות האקטביציה.

הנתונים גם כאן מנורמלים על ידי ממוצע וסטיית תקן, והאימון של המודל נעשה ב-50 איפוקים.