

שם: אליעד ארזואן

ת.ז: 206482622

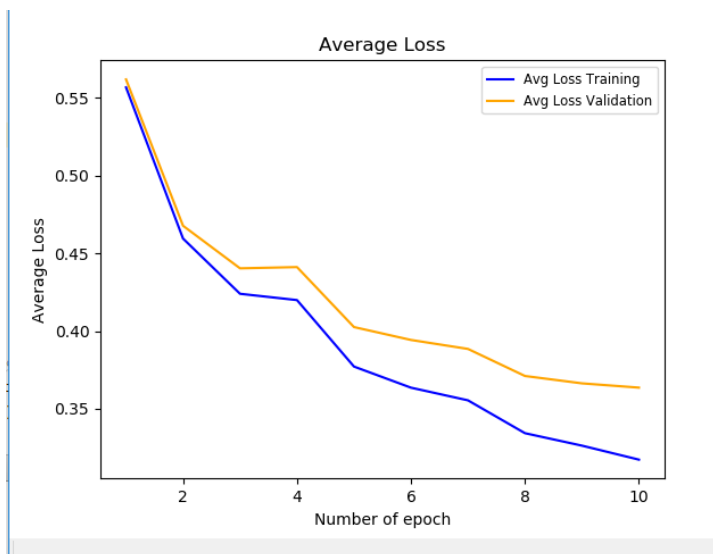
### תרגיל 4 למידת מכונה – הסבר עבודה בסוף

עבור כל מודל נראה את התוצאות שמקבלים באיפוק האחרון ואת הגרף של התוצאות של כל איפוק. לכן נראה את התוצאות באיפוק העשירי ואת כל האיפוקים בגרף.

(a) מודל a עם שני hidden layers שעל כל שכבה מופעלת פונקציה Relu:

- Training set: **Average loss**: 0.3172, **Accuracy**: 42544/48000 (88%)
- Validation set: **Average loss**: 0.3636, **Accuracy**: 10434/12000 (86%)
- Test set: **Average loss**: 0.3820, **Accuracy**: 8609/10032 (85%)

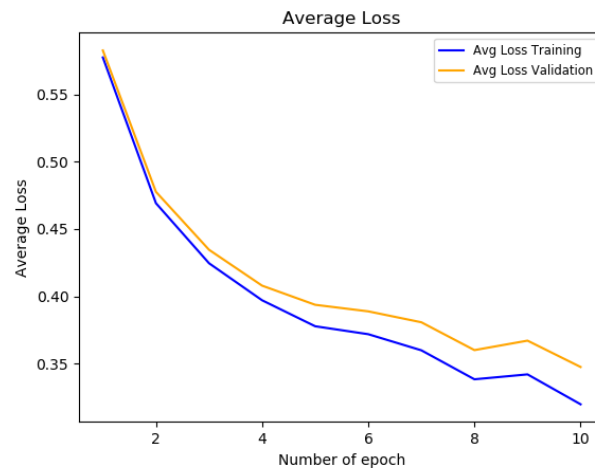
גרף של loss הממוצע לעומת האיפוק:



(b) מודל b עם הוספת שכבות dropout למודל a:

הוספנו שכבות dropout אחרי הפלט של הhidden layers.

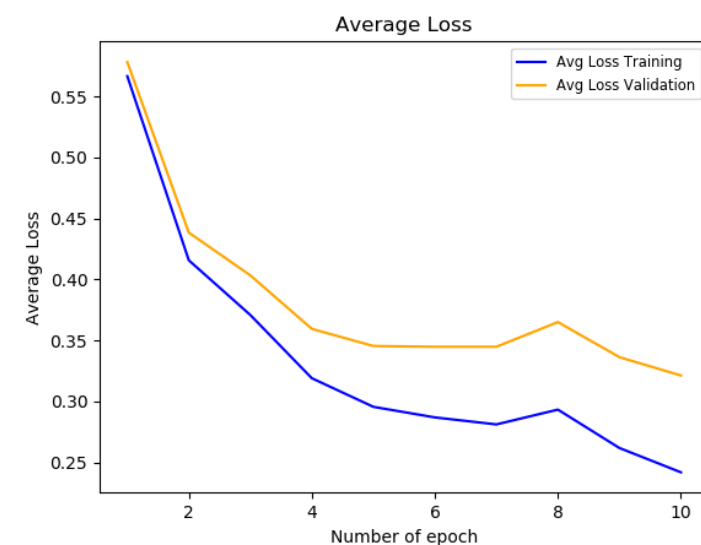
- Training set: **Average loss:** 0.3198, **Accuracy:** 42432/48000 (88%)
- Validation set: **Average loss:** 0.3475, **Accuracy:** 10508/12000 (87%)
- Test set: **Average loss:** 0.3758, **Accuracy:** 8655/10032 (86%)



(c) מודל c – הוספת batch normalization:

הפעלנו batch לפני פונקציות האקטיבציה

- Training set: **Average loss:** 0.2420, **Accuracy:** 43795/48000 (91%)
- Validation set: **Average loss:** 0.3214, **Accuracy:** 10565/12000 (88%)
- Test set: **Average loss:** 0.3446, **Accuracy:** 8733/10032 (87%)



## הסבר עבודה:

- דבר ראשון טענו את mnistn שלנו באמצעות החבילות של torch. 89
- חלקנו את הסט שטענו ל80% של אימון ו20% של ולידציה.
- בסופו של דבר בחנו על הטסט.

השתמשנו בשלושה מודלים כאשר כל אחד מהם יורש מnn.module.

**המודל הראשון-** רשת נוירונים עם 2 שכבות כך שעל כל שכבה מופעלת פונקציית relu ובסוף נעשה softmax ונקח את ההסתברות המקסימלית.

**המודל השני-** כמו המודל הראשון רק ש"סינון נוסף" שלנו יהיה dropout. נשתמש בכלים של pytorch לביצוע זאת – ניסיתי כמה הסתברויות האמת ש0.1 ו0.35 נתנו לי תוצאות טובות.

**המודל השלישי-** כמו המודל הראשון רק שנוסיף לפני פונקציות האקטיבציה בhidden layers את batch normalization.