

## תרגיל 4

### רשת נוירונים עם 2 hidden layer:

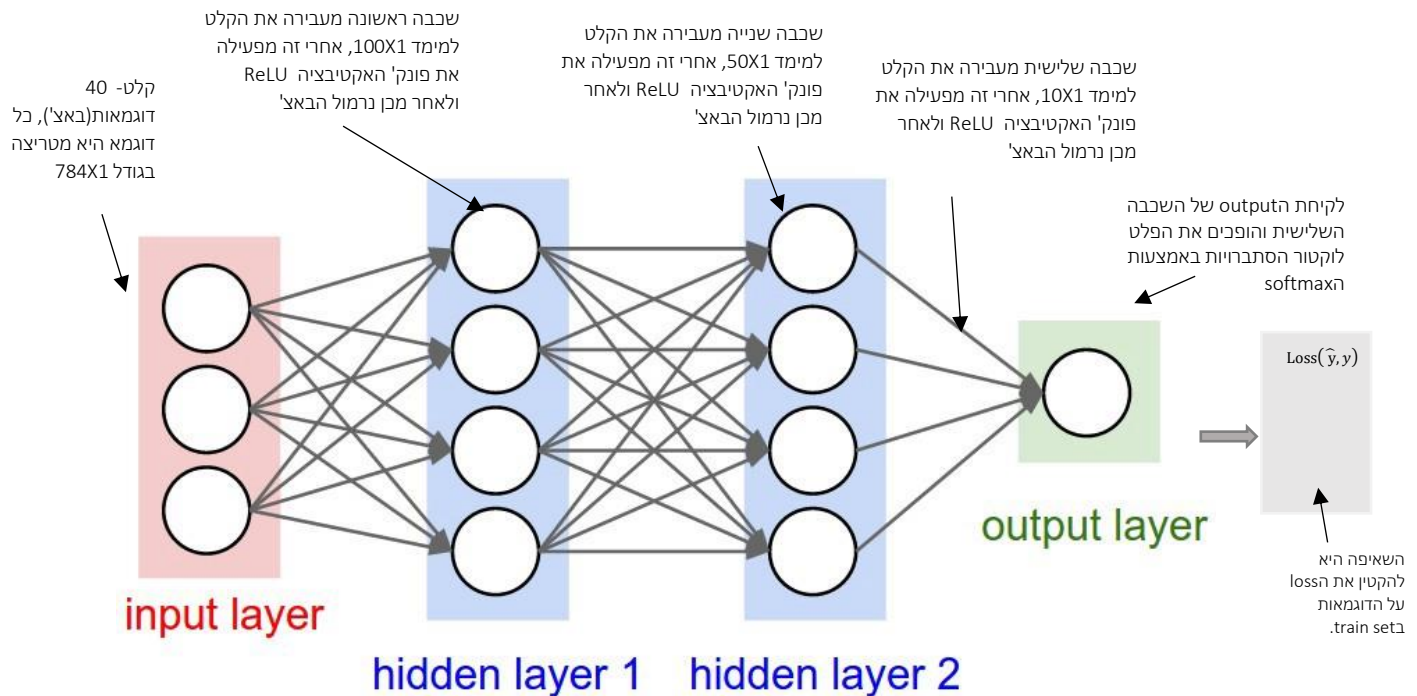
המודל מכיל את הפרמטרים הלומדים משכבות הנוירונים  $(W1, W2, W3, b1, b2, b3)$ .

המודל שלי כולל את ההיפר פרמטרים (לא לומדים) הבאים:

Hidden layer size, epochs, activation function and batch size.

בחרתי להשתמש בעדכון הפרמטרים לפי AdaGrad כיוון שעבור מספר ה epochs הקטן יחסית שהתבקשנו עבורו, ה AdaGrad מחזק את ה features הפחות שכיחים עם learn rate גבוה יחסית לעומת ה features השכיחים תוך כדי שהמודל כן לומד בצורה יחסית טובה על הפיצ'רים השכיחים (מכיוון שמספר ה epochs קטן יחסית).

המודל עובד לפי השרטוט הבא:



את הפרמטרים אימנתי במהלך פונקציית ה training במעבר על test, ה optimizer לפי שיטת Adagrad (מפורט למעלה).

כעת אפרט על כל היפר-פרמטר ומדוע בחרתי לאמן את המודל שלי כפי שבחרתי לאמנו:

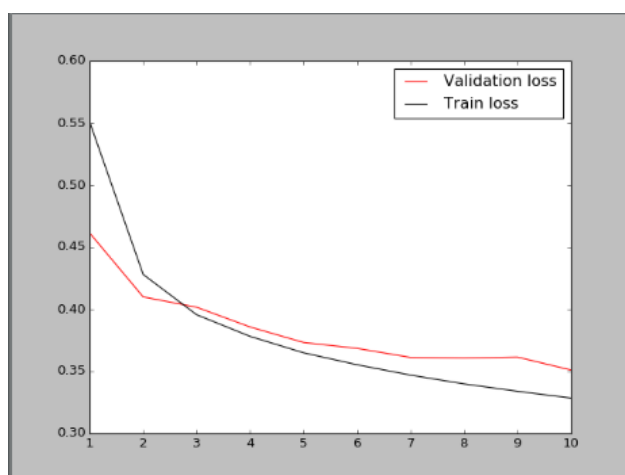
- 1) Batch size: השתמשתי ב batch size 40 כ input למודל. נרמלתי את ה input (batch normalization), הנחתי שכיוון שיש לנו מספר epochs קטן יחסית אז נרמול ה input יאפשר למידה "אחידה" יותר עליו. נרמול ה input מצמצם את "קיצוניות" המידע.
- 2) Hidden layer size: כדי שהמודל יהיה אפקטיבי, נרצה לצמצם את הבעיה (מעבר מבעיה בגודל 784 לגודל קטן יותר) באופן הדרגתי, גודל ה hidden layer חייב להיות בין מספר הקלאסים לגודל הקלט. התבקשנו לגודל 100 בשכבה הראשונה, גודל 50 בשכבה השנייה.
- 3) Learn rate ( $\eta = \mu$ ): עדכון הפרמטרים הלומדים, נשים לב שב Adagrad אין learn rate מוגדר מראש, וככל ש feature יותר שכיח כך ה learn rate שלו קטן (כי מחלקים בסכום הגרדיאנטים) ולכן ה learn rate יקבע באופן דינמי תוך כדי הריצה.
- 4) Epochs: מספר הפעמים שעוברים על ה training set. התבקשנו לעשות 10 epochs.
- 5) Activate function: פונקציית האקטיבציה מהאפשרויות הבאות Sigmoid, ReLU, PReLU, tanh ועוד. מטרת פונקציית האקטיבציה היא הכנסת אלמנט אי לינאריות במודל במעבר בין השכבות. כאן התבקשנו להשתמש בפונקציית ReLU.

### תוצאות המודלים:

**מודל a:** רשת נוירונים עם 2 hidden layers עם optimizer מסוג Adagrad ועם פונקציית אקטיבציה ReLU:

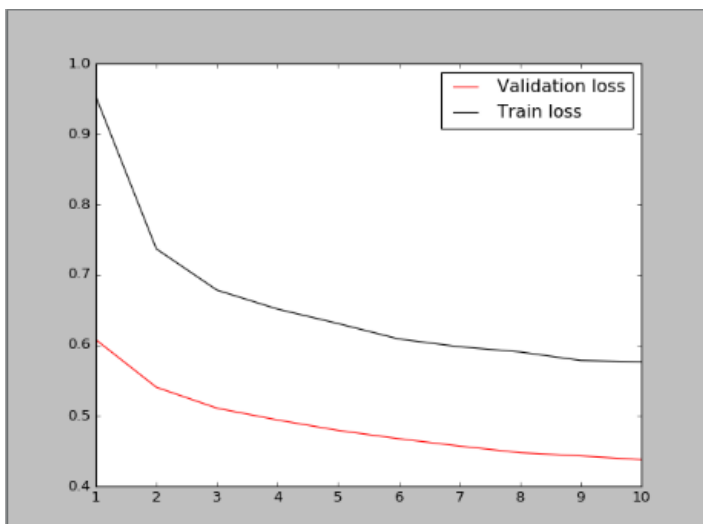
(81%) Train epoch number: 1, Average loss: 0.5507, Accuracy: 38732/48000  
(84%) Validation epoch number :1 Average loss: 0.4614, Accuracy: 10047/12000  
(85%) Train epoch number: 2, Average loss: 0.4281, Accuracy: 40781/48000  
(86%) Validation epoch number :2 Average loss: 0.4100, Accuracy: 10268/12000  
(86%) Train epoch number: 3, Average loss: 0.3956, Accuracy: 41296/48000  
(86%) Validation epoch number :3 Average loss: 0.4017, Accuracy: 10272/12000  
(87%) Train epoch number: 4, Average loss: 0.3782, Accuracy: 41620/48000  
(86%) Validation epoch number :4 Average loss: 0.3857, Accuracy: 10346/12000  
(87%) Train epoch number: 5, Average loss: 0.3650, Accuracy: 41789/48000  
(87%) Validation epoch number :5 Average loss: 0.3733, Accuracy: 10393/12000  
(87%) Train epoch number: 6, Average loss: 0.3553, Accuracy: 41897/48000  
(87%) Validation epoch number :6 Average loss: 0.3685, Accuracy: 10419/12000  
(88%) Train epoch number: 7, Average loss: 0.3470, Accuracy: 42089/48000  
(87%) Validation epoch number :7 Average loss: 0.3612, Accuracy: 10436/12000  
(88%) Train epoch number: 8, Average loss: 0.3399, Accuracy: 42177/48000  
(87%) Validation epoch number :8 Average loss: 0.3607, Accuracy: 10455/12000  
(88%) Train epoch number: 9, Average loss: 0.3340, Accuracy: 42298/48000  
(87%) Validation epoch number :9 Average loss: 0.3615, Accuracy: 10430/12000  
(88%) Train epoch number: 10, Average loss: 0.3285, Accuracy: 42410/48000  
(87%) Validation epoch number :10 Average loss: 0.3511, Accuracy: 10488/12000

(86%) Test Average loss: 0.3809, Accuracy: 8620/10000



**מודל ב:** רשת נוירונים עם 2 hidden layers, עם optimizer מסוג Adagrad ועם פונקציית אקטיבציה ReLU ובנוסף dropout עם  $p = 0.5$  מאפשר לנו לגרום לרשת הנוירונים לא לנטות ל overfitting ע"י כיבוי נוירונים בהסתברות 0.5, כך גם ניתן לאמן המון רשתות ( $2^n$  אם יש הסתברות 0.5) (בכל רשת יש נוירונים אחרים שלא כבויים). לא הצלחתי לגרום למודל להוציא תוצאות טובות בשימוש dropout (ייתכן כי בעזרת 10 epochs לא ניתן לאמן עם dropout בצורה טובה)

(64%) Train epoch number: 1, Average loss: 0.9535, Accuracy: 30601/48000  
 (78%) Validation epoch number :1 Average loss: 0.6081, Accuracy: 9367/12000  
 (73%) Train epoch number: 2, Average loss: 0.7369, Accuracy: 35080/48000  
 (80%) Validation epoch number :2 Average loss: 0.5405, Accuracy: 9646/12000  
 (76%) Train epoch number: 3, Average loss: 0.6784, Accuracy: 36429/48000  
 (82%) Validation epoch number :3 Average loss: 0.5107, Accuracy: 9810/12000  
 (77%) Train epoch number: 4, Average loss: 0.6514, Accuracy: 36932/48000  
 (82%) Validation epoch number :4 Average loss: 0.4939, Accuracy: 9872/12000  
 (78%) Train epoch number: 5, Average loss: 0.6308, Accuracy: 37343/48000  
 (83%) Validation epoch number :5 Average loss: 0.4792, Accuracy: 9923/12000  
 (79%) Train epoch number: 6, Average loss: 0.6089, Accuracy: 37781/48000  
 (83%) Validation epoch number :6 Average loss: 0.4673, Accuracy: 9983/12000  
 (79%) Train epoch number: 7, Average loss: 0.5980, Accuracy: 38097/48000  
 (83%) Validation epoch number :7 Average loss: 0.4569, Accuracy: 10019/12000  
 (80%) Train epoch number: 8, Average loss: 0.5905, Accuracy: 38245/48000  
 (84%) Validation epoch number :8 Average loss: 0.4475, Accuracy: 10084/12000  
 (80%) Train epoch number: 9, Average loss: 0.5785, Accuracy: 38365/48000  
 (84%) Validation epoch number :9 Average loss: 0.4429, Accuracy: 10098/12000  
 (80%) Train epoch number: 10, Average loss: 0.5766, Accuracy: 38523/48000  
 (84%) Validation epoch number :10 Average loss: 0.4376, Accuracy: 10121/12000



(83%) Test Average loss: 0.4546, Accuracy: 8331/10000

**מודל c:** רשת נוירונים עם 2 hidden layers, עם optimizer מסוג Adagrad ועם פונקציית אקטיבציה ReLU ובנוסף batch normalization על קבוצת דוגמאות בגודל 40, הסברתי בעמוד הראשון (מודל זה הוא המודל שנתן לי את התוצאות הכי טובות ולכן בחרתי בו) על ה batch normalization וה batch size.

(83%) Train epoch number: 1, Average loss: 0.5042, Accuracy: 39765/48000  
(87%) Validation epoch number :1 Average loss: 0.3835, Accuracy: 10388/12000  
(86%) Train epoch number: 2, Average loss: 0.3804, Accuracy: 41481/48000  
(88%) Validation epoch number :2 Average loss: 0.3493, Accuracy: 10503/12000  
(87%) Train epoch number: 3, Average loss: 0.3481, Accuracy: 41943/48000  
(88%) Validation epoch number :3 Average loss: 0.3434, Accuracy: 10550/12000  
(88%) Train epoch number: 4, Average loss: 0.3316, Accuracy: 42307/48000  
(88%) Validation epoch number :4 Average loss: 0.3388, Accuracy: 10540/12000  
(89%) Train epoch number: 5, Average loss: 0.3143, Accuracy: 42621/48000  
(89%) Validation epoch number :5 Average loss: 0.3254, Accuracy: 10628/12000  
(89%) Train epoch number: 6, Average loss: 0.3045, Accuracy: 42715/48000  
(89%) Validation epoch number :6 Average loss: 0.3203, Accuracy: 10625/12000  
(89%) Train epoch number: 7, Average loss: 0.2941, Accuracy: 42854/48000  
(89%) Validation epoch number :7 Average loss: 0.3163, Accuracy: 10652/12000  
(90%) Train epoch number: 8, Average loss: 0.2868, Accuracy: 42970/48000  
(89%) Validation epoch number :8 Average loss: 0.3103, Accuracy: 10679/12000  
(90%) Train epoch number: 9, Average loss: 0.2794, Accuracy: 43002/48000  
(89%) Validation epoch number :9 Average loss: 0.3091, Accuracy: 10656/12000  
(90%) Train epoch number: 10, Average loss: 0.2738, Accuracy: 43151/48000  
(89%) Validation epoch number :10 Average loss: 0.3127, Accuracy: 10681/12000  
  
(88%) Test Average loss: 0.3433, Accuracy: 8792/10000

