

עבודה מספר 2:

<https://github.com/mooshon1989/DL/tree/master/hw2>

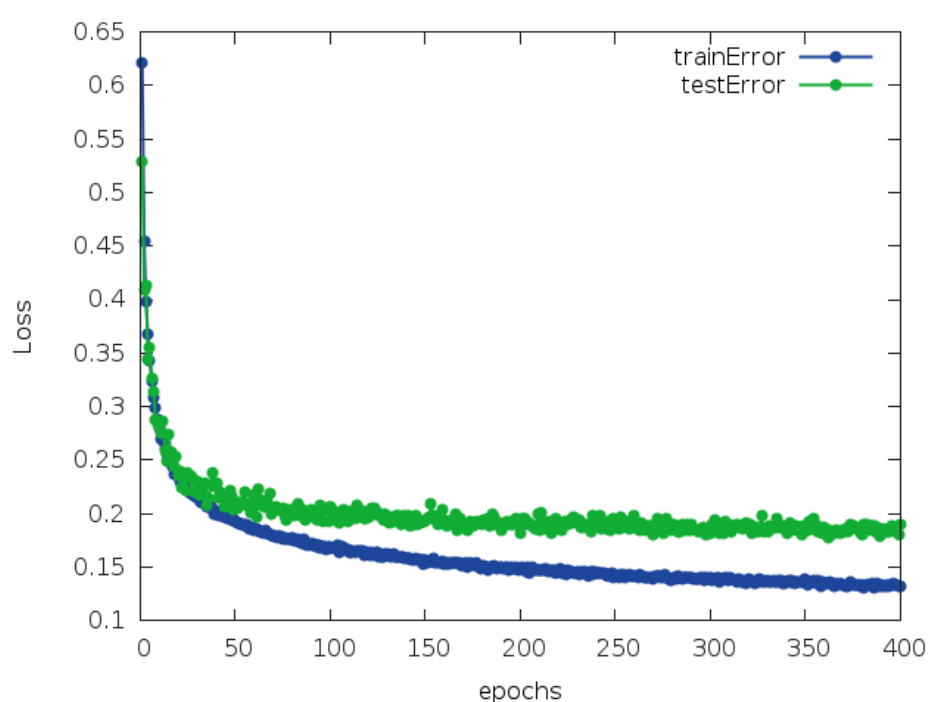
1. תיאור המודל:

בנינו מודל על בסיס רשת NN עם 48,276 משתנים (במודל הטוב ביותר שיצרנו). תחילה ביצענו נרמול של הנתונים סביב התוחלת עפ"י נתוני האימון. לאחר מכן מימשנו פונקציה- BatchFlip אשר מכילה את מודל ה- data augmentation. הרשת בנויה ממספר שכבות הנוצרות ע"י קריאה לפונקציית Block המכילה: שכבת קונבולוציה (SpatialConvolution), שכבת נורמליזציה (SpatialBatchNormalization) ופונקציית אקטיבציה (ReLU). כמו כן, בין הקריאות לפונקציה block הפעלנו: SpatialMaxPooling ו- SpatialAveragePooling.

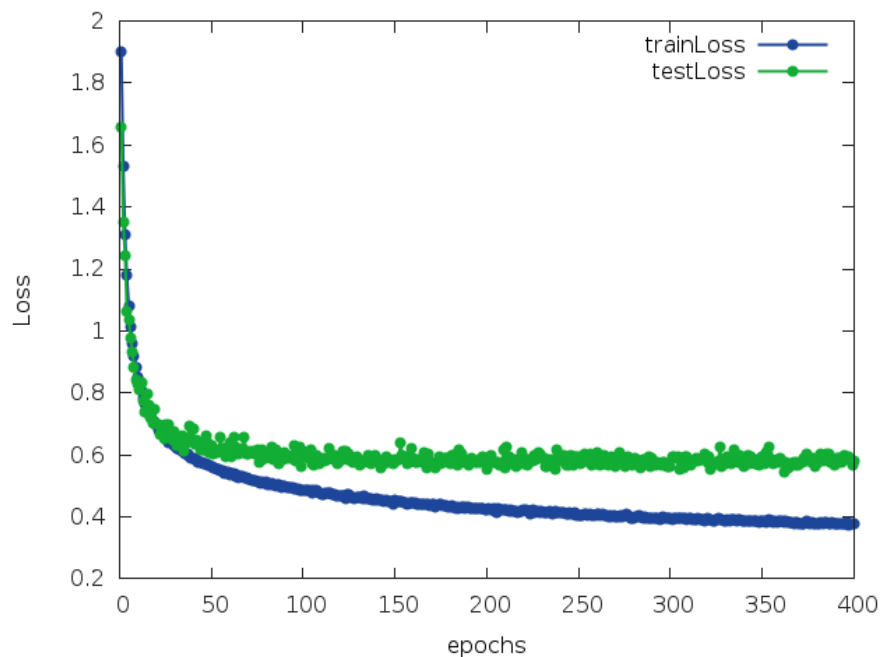
2. המודל הסופי:

שימוש בפונקציית **hflip** (data augmentation) ובפונקציית אופטימיזציה **adam**.

2.1. גרף 1- שגיאת המבחן והאימון כתלות במספר ה- Epochs:



2.2. גרף 2- ערך פונקציית ה- loss של נתוני המבחן והאימון כתלות במספר ה- Epochs :



2.3. מסקנות :

הדיוק הטוב ביותר שקיבלנו: 82.021233974359% .

מספר ה- epochs עבור דיוק זה: 400 (הייתה התייצבות אחרי 240 epochs).

ניתן לראות מהגרף 1 כי קיים קשר הפוך בין מספר ה- Epochs לשגיאת האימון, כלומר ככל שמספר ה- Epochs עולה כך שגיאת האימון יורדת. בנוסף, ניתן לראות שאם לא משתמשים כלל בחלוקה ל- Epochs שגיאת האימון הינה 0.655 בקירוב ויורדת באופן די קיצוני ל- 0.3 אם משתמשים בין 10-2 Epochs. כמו כן, החל מ- Epochs 250 קיימת התכנסות ל- 0.15 בשגיאת האימון.

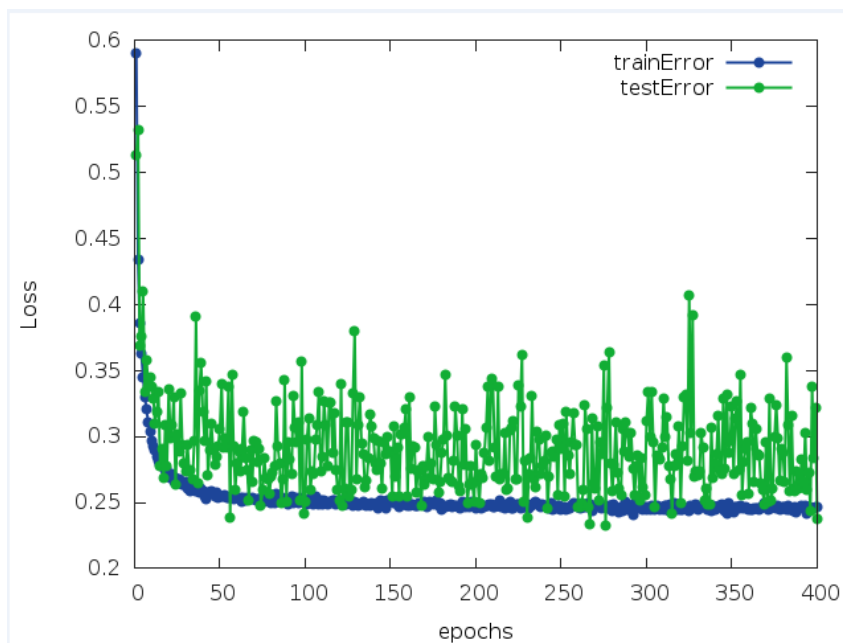
לגבי שגיאת המבחן, החל מ- Epochs 100 קיימת התכנסות סביב 0.2-0.25 בשגיאת האימון, כלומר מודל שנבנה עם חלוקה ל- Epochs 100 ומודל שנבנה עם חלוקה ל- 400 Epochs יניבו אותה שגיאה בקירוב. בנוסף, ניתן לראות 'רעשים' סביב Epochs 50.

ניתן לראות מגרף 2, שקיים קשר הפוך בין מספר ה- Epochs לערך פונקציית ה- trainLoss. באופן דומה לגרף 1, יש ירידה די קיצונית בערך פונקציית ה- trainLoss אם משתמשים ב- Epoch אחד לעומת שימוש במספר Epochs בודדים. החל מ- 200 Epochs ערך פונקציית ה- trainLoss מתכנסת ל- 0.4. לגבי ערך פונקציית ה- testLoss, ניתן לראות שהחל מ- Epochs 50 קיימת התכנסות סביב 0.6-0.7.

3. מודלים נוספים:

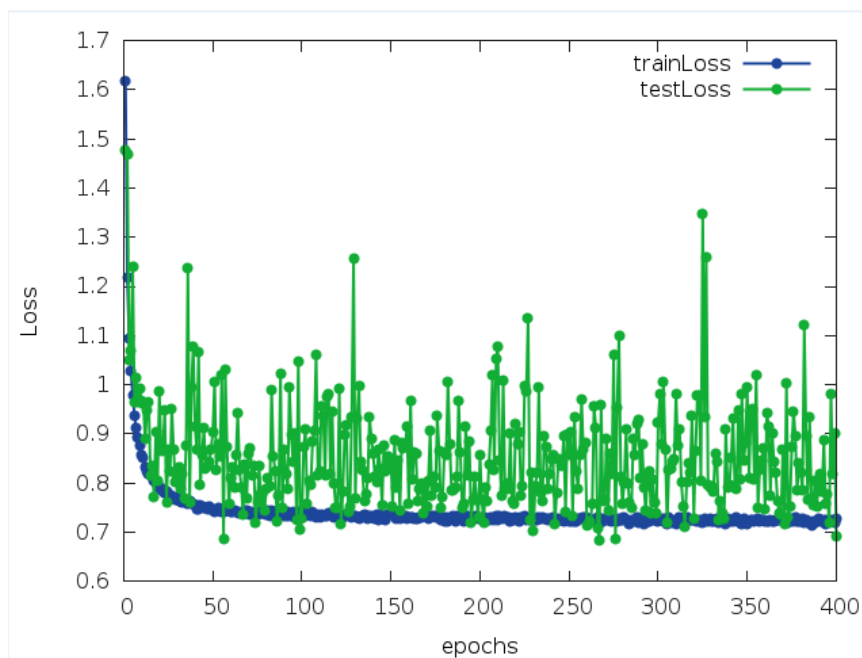
3.1. שימוש בפונקציית **hflip** (data augmentation) ובפונקציית אופטימיזציה **sgd**.

3.1.1. גרף 3- שגיאת המבחן והאימון כתלות במספר ה- Epochs:



3.1.2. גרף 4- ערך פונקציית ה- loss של נתוני המבחן והאימון כתלות במספר ה-

Epochs:



3.1.3. מסקנות:

הדיוק שקיבלנו: 76.252003205128%.

מספר ה- epochs עבור דיוק זה: 400.

ניתן לראות מהגרף 3 כי קיים קשר הפוך בין מספר ה- Epochs לשגיאת האימון. כמו כן, החל מ- Epochs 50 קיימת התכנסות ל- 0.25 בשגיאת האימון (שגיאה גבוהה יותר המודל הסופי שבו ההתכנסות היא ל- 0.15 בשגיאת האימון).

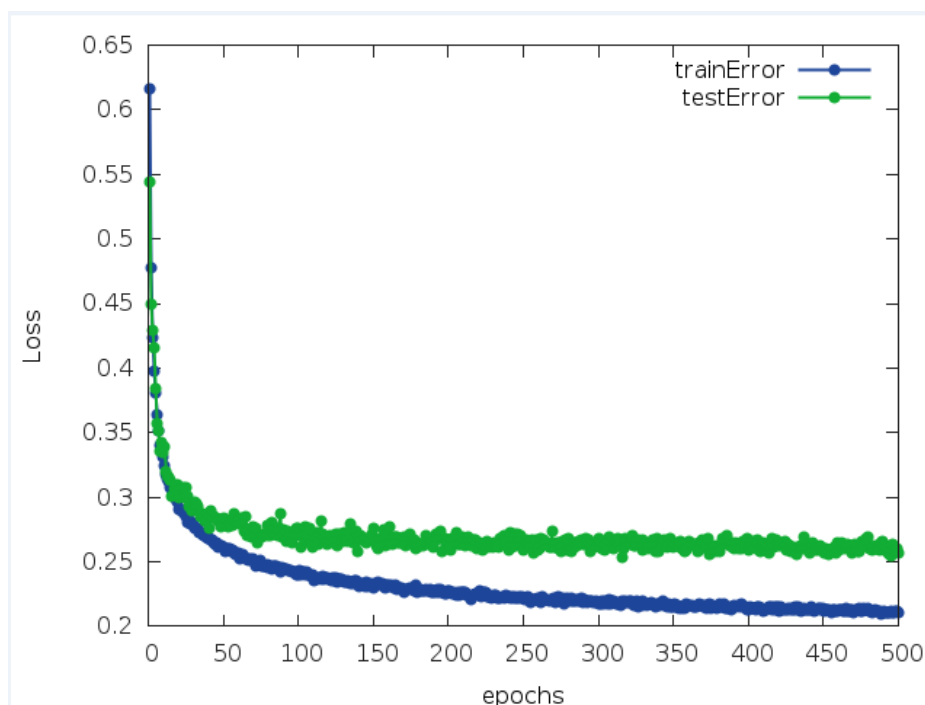
ניתן לראות מגרף 4 כי החל מ- Epochs 50 ערך פונקציית ה- trainLoss מתכנסת ל- 0.7, (ערך גבוה מהמודל הסופי שבו ההתכנסות היא ל- 0.4 בערך פונקציית ה- trainLoss).

משני הגרפים על נתוני המבחן (עקומה בצבע ירוק) ניתן לראות כי אין יציבות גם בשגיאת המבחן וגם בערך פונקציית ה- testLoss (אין התכנסות – 'קפיצות' בגרף). כלומר ניתן להסיק כי בהרצה עם מספר epochs גדול יותר הייתה בסופו של דבר התכנסות לערך מסוים.

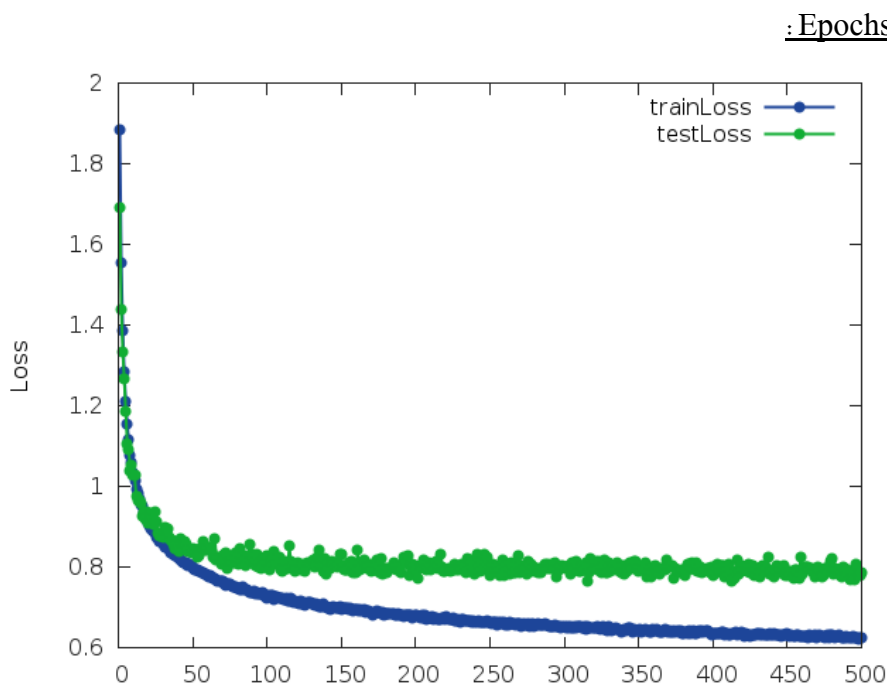
3.2. שימוש בקומבינציה של פונקציית **randomcrop** : 'reflection' ו- 'zero' (data)

augmentation) ופונקציית אופטימיזציה **adam**.

3.2.1. גרף 5- שגיאת המבחן והאימון כתלות במספר ה- Epochs :



3.2.2. גרף 6 - ערך פונקציית ה-loss של נתוני המבחן והאימון כתלות במספר ה-



3.2.3. מסקנות :

הדיוק שקיבלנו: 80.829326923077%.

מספר ה- epochs עבור דיוק זה: 400.

ניתן לראות מהגרף 5 כי קיים קשר הפוך בין מספר ה- Epochs לשגיאת האימון. כמו כן, החל מ- Epochs 50 קיימת ירידה מתונה עד ל- 0.2 בשגיאת האימון (שגיאה גבוהה יותר המודל הסופי שבו ההתכנסות היא ל- 0.15 בשגיאת האימון). לגבי שגיאת המבחן, החל מ- Epochs 100 קיימת התכנסות סביב 0.25-0.3 בשגיאת האימון (טווח גדול יותר מהמודל הסופי שבו ההתכנסות היא סביב 0.2-0.25).

ניתן לראות מגרף 4 כי החל מ- Epochs 50 קיימת ירידה מתונה בערך פונקציית ה- trainLoss עד ל-0.6, (ערך גבוה מהמודל הסופי שבו ההתכנסות היא ל- 0.4 בערך פונקציית ה- trainLoss). לגבי ערך פונקציית ה- testLoss, ניתן לראות שהחל מ- Epochs 50 קיימת התכנסות סביב 0.8-0.9 (טווח גדול יותר מהמודל הסופי שבו ההתכנסות היא סביב 0.6-0.7).