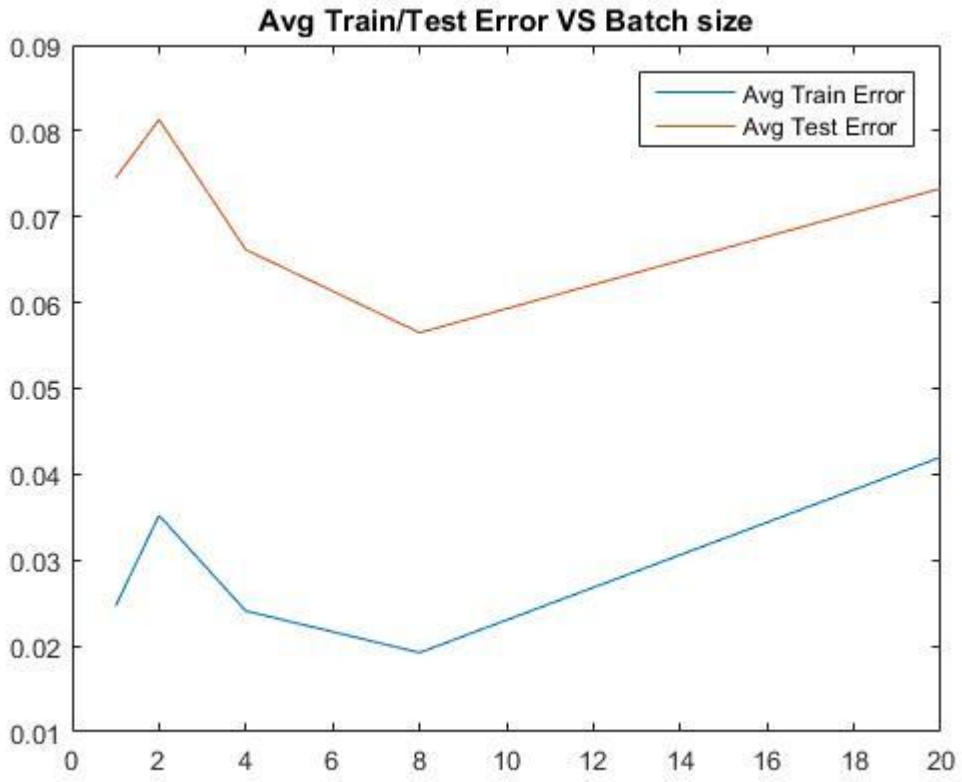
Report for exercise 4

Part1:

הגרף של שגיאת אימון ממוצעת כפונק' של גודל הבאץ':



רואים ששגיאת האימון האופטימלי מתקבלת עבור באץ' בגודל 8.

בדיוק של חישוב הגרדיאנט בכל איטרציה תלוי בגודל הבאץ' , ככל שגודל הבאץ' גדל ככל שחישוב הגרדיאנט הוא מדוייק יותר.

הטרייד-אוף הוא בין דיוק חישוב הגרדיאנט לבין מספר האיטריציות.

רואים מהגרף שככל גדל גודל הבאץ' (אחרי גודל 8) דווקא ממוצע שגיאת האמון גדלה וזה נ ובע מכך שגדילת גודל הבאצ' יותר מדי מביאה למספר איטרציות קטן של האלגוריתם מה שפוגע בדיוק חישוב הגרדיאנט.

נסתכל גם על גודל מדגם האימון, כאשר הוא גדל אז צריך לחשב את הגרדיאנט יותר הרבה ממדגם אמון קטן ולכן זה צריך הרבה איטרציות ולכן נצטרך גודל באצ' מתאים (לא קטן אבל אופטימלי) כי לא נוכל לומר קטן כרצוננו כי בגרף רואים שהשגיאה האופטימלית התקבל בערך 8 ולא 2.

Part 2:

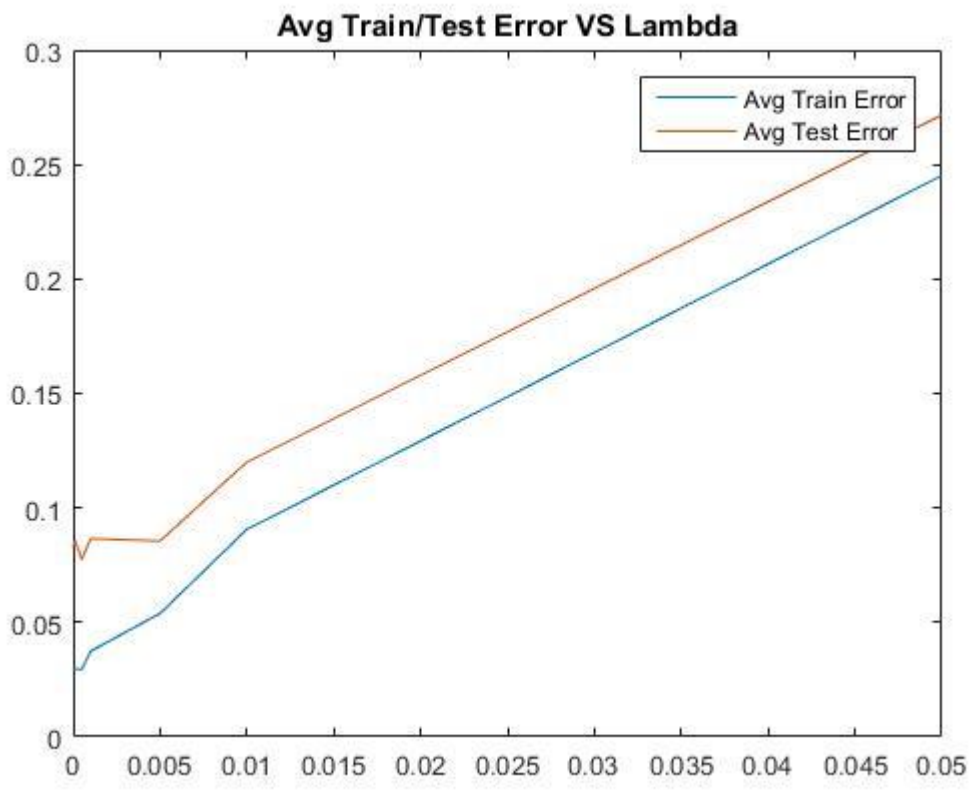
Momentum variable **0.9** with best learning rate of **0.001** get **0.0218** **train error**

Momentum variable **0.95** with best learning rate of **0.001** get **0.0188** **train error**

Momentum variable **0.99** with best learning rate of **0.0001** get **0.0268** **train error**

ערכים גבוהים של מומנטים מגדילים את הפער בין העדכון הקודם לבין העדכון הנוכחי, ולכן משתמשים ב- לירנינג רייטס נמוכים כדי להתגבר על פער זה ולאזן.

Part 3:



רואים שכאשר למבדה גדל השגיאות של סט האימון והבחינה גדלים, ולהפך , זה דבר תלוי בקשר עם גודל מחלקת ההיפותיזות שכאשר למדבה גדל מחלקת ההיפתוזיות קטנה ולהפך.

הגרף שונה מלימוד קלאסי בכך ששם גרף סט הבחינה היה דומה לפרפולה.

נשים לב שלא ברור מהגרפים אם למבדה קטן מאוד אז יש OVERFIITING, גם שם היינו מזהים את התופעה יותר ברור.

Part 4:

תוצאות:

Average Train Error: 0.891

Average Test Err r : 0.900

רואים ששגיאת האימון והטסט מאוד גדולים.

כאן הכפלנו את הנגזרת החלקית באפס כאשר מאתחילים את המשקל של השכבה באפס כלומר הגרדינט שמתקבל הוא אפס. זה לא קורה בלמידה קלאסית כי שם הקטנת הגרדיאנט עד לאפס מביא אותנו למינימום של הפונ' כאשר הפונק' קמורה.

Part 5:

תוצאות:

Average Train Error: 0.010

Average Test Error: 0.060

קבלנו שגיאות מאוד קטנות , משום שאחרי 7500 איטרציות מתוך 10,000 היינו מאוד קרובים למינימום וצעדים של התקרבות קטנים היו ממש מביאים אותנו למינימום ולכן הקטנת הלילקנינג רייטס נתנתה תוצאה טובה יותר.

Part 6:

תוצאות:

Train Error : 0

Test Erro : 0.0403

```
batch_size    = 8
T      = 1e4;          % No. of iterations (10K x 1-batch = 10K examples
passed through the network)
mu     = single(0.8);    % Momentum variable
```

```
lambda = single(0.0005);    % Regularization constant
eta    = @(t)(0.05);
```

אני לקחתי תוצאות של סעיפים קודמים והתחלתי לנסות וזו התוצאות הכי טובות שקבלתי.

לקחת הבאצ' הכי טוב מסעיף אל והלמבדה הכי טובה מסעיף 3 לפי הגרף ואיטא 0.05.

Part 7:

תוצאות:

Train Error: 0

Test Error: 0.0307

רשת גדולה יותר גוררת למחלקת היפותיזות עשירה יותר , וזה גורר לשגיאת הטסט להיות נמוכה יותר.

כיוון שמחלקת ההיפותיזות עשירה יותר זאת אומרת שהביאס קטן יותר כי התוצאות יותר מדויקות, אבל מציאת התוצאה הכי טובה בין כל התוצאות זה קשה יורתר לכן הווריאנס גדל, לכן רואים טרייד-אוף בין הביאס לווריאנס.

Part 8:

תוצאות:

Train Error : 0

Test Error: 0.0248

כאן גם הסתמכתי על סעיפים קודמים וכל ערכי הפרמטרים שמזערו את ערך השגיאה והגעתי לעכרים אלה:

```
batch_size    = 10;
conv_kernel   = 5;
conv_stride   = 1;
conv_channels = 20 and then changed to 50 for one layer;
pool_kernel   = 2;
pool_stride   = 2;
H = 28;      % Image height
W = 28;      % Image width
B = 1;       % # of bands (grayscale)
k = 10 and then changed to 500 for one layer;
T      = 1e4;
mu     = single(0.8);    % Momentum variable
lambda = single(0.0005);
```

eta = 0.05 for the first 7500 iteration and then changed to 0.005

מה שמוזר הוא שעבור גודל באץ' 8 קבלתי שגיאה של הבחינה גדול מגודל באץ' 10 ולכן שמתי באצ' 10 אף על פי שבסעיף א קבלנו שהאופטימלי הוא באץ' 8 ולא 10.

Bonus Part: