

חישוביות וקוגניציה - תרגיל 4

להגשה עד: 24/11/2021

שימו לב: שאלה 1 היא שאלה אנליטית ושאלה 2 היא שאלת תכנות

שאלה 1

נניח שברצוננו לאמן רשת נוירונים Feed-Forward בעזרת למידת גרדיאנט, לפי אלגוריתם BackPropagation.

1. א. כרגיל באלגוריתמים של למידת גרדיאנט, ראשית עלינו לאתחל את המשקולות ברשת (מטריצות הקשרים $\mathbf{W}^1, \dots, \mathbf{W}^l$). נניח שמטריצות הקשרים מאותחלות כולן ל 0 (כלומר $W_{ij}^l = 0$ לכל l, i, j), ושפונקציית האקטיבציה ברשת מקיימת $g(0) = 0$. מה תהיה ההשלכה לגבי הלמידה ברשת?

רשות: מה קורה אם $g(0) \neq 0$?

ב. מה יקרה אם נאתחל את כל המשקולות באותה השכבה לערך קבוע כלשהו $a \neq 0$?

2. בתרגיל 1 בנייתם רשת רב-שכבתית של פרספטרונים בינאריים כדי לממש כלל-החלטה שפרספטרון בודד לא היה יכול לממש. נניח שבמקום למצוא את המשקולות 'באופן ידני', היינו רוצים לקבוע רק את הארכיטקטורה וללמוד את המשקולות עצמם (למידה מפוקחת בעזרת דוגמאות). האם ניתן להשתמש באלגוריתם Back Propagation כדי לאמן רשת רב-שכבתית כזו של פרספטרונים בינאריים? הסבירו.

הערה: שאלות 1 ו 2 הן בלתי-תלויות

שאלה 2

בשאלה זו תאמנו רשת נוירונים שתלמד לזהות ספרות בכתב יד. לצורך כך נשתמש ב data set מפורסם בשם MNIST. הדוגמאות הן תמונות (בגוויי אפור) בגודל 28×28 , ואנחנו נתייחס אליהן בתור וקטורים במימד $28^2 = 784$. ערכי התמונות נורמלו להיות בין 0 (שחור) ל 1 (לבן). לכל תמונה יש תיוג מתאים של אחת מ-10 הספרות האפשריות $0, 1, \dots, 9$.

הפונקציית המצורפת loadMNISTImages ו-loadMNISTLabels טוענות את הדוגמאות והתיוגים. שימו לב שהמידע מחולק לשניים: training set ובו 60,000 דוגמאות, ו test set שבו 10,000 דוגמאות נוספות. האלגוריתם שתריצו ילמד את הדוגמאות ב-training set, ואחר כך נבדוק את שגיאת ההכללה שלו על הדוגמאות ב-test set.

ארכיטקטורה

נרצה לאמן רשת נוירונים רב שכבתית, כאשר פונקציית האקטיבציה של הנוירונים היא $g \equiv \tanh$. שכבת הקלט ברשת שתאמנו היא מגודל 784 ושכבת הפלט מגודל 10. בתור תיוג נשתמש בוקטורי בסיס סטנדרטיים (one hot encoding) של הספרות. למשל, אם התיוג של תמונה הוא $y = 5$ אז 'הפלט הרצוי' של הרשת יהיה וקטור באורך 10 שכולו אפסים חוץ מהמקום ה-6 שיהיה 1 (שימו לב שהספרה הראשונה היא 0 ולא 1). פונקציית השגיאה היא ריבועית, כלומר חצי המרחק האוקלידי בין וקטור הפעילות בשכבת הקלט לבין וקטור התיוג המתאים: $\varepsilon = \frac{1}{2}(\mathbf{s}_1^M - \hat{\mathbf{y}})^2$. שימו לב, הפונקציית loadMNISTLabels מחזירה את התיוגים בפורמט הזה (כלומר מטריצה בגודל 10 על מספר הדוגמאות).

בזמן testing, ע'מ לחשב את דיוק הרשת (כמה מהדוגמאות ב test set הרשת מתייגת נכונה) נגדיר את התיוג שהרשת מחשבת לפי הנוירון בעל האקטיבציה המקסימלית בשכבת הפלט (למשל, אם הנוירון ה'פעיל' ביותר הוא נוירון 3, אז נאמר שהרשת מתייגת את התמונה כספרה 2).

מימוש

השתמשו במחלקה המצורפת $1FF$ אשר מכילה מימוש חלקי לרשת נוירונים רב שכבתית. עליכם להשלים את הפונקציות:

1. `loss_deriv`: הפונקציה מקבלת את ערכי הפעילות בשכבת הפלט ואת התיג הנכון, ומחזירה את נגזרת השגיאה ביחס לפעילות בשכבת הפלט. כלומר, מחשבת את $\nabla_{s_1^M}(\varepsilon(s_1^M, \hat{y}))$.
2. `backprop`: הפונקציה מקבלת סט של דוגמאות (mini-batch) והתיג המתאים להם ומחזירה את הנגזרות החלקיות של השגיאה לפי כל אחת ממטריצות המשקולות (שימו לב - אין לעדכן את המשקולות, רק להחזיר את הנגזרות המתאימות)

הערות למימוש סעיף 2:

- העזרו בפונקציה `loss_deriv` שמימשתם וכן בפונקציות `activation`, `activation_deriv` הנתונות
- לצורך יעילות, כתבו את המימוש בצורה וקטורית\מטריציונית, בלי לולאות על נוירונים בתוך כל שכבה ובלי לולאות על פני הדוגמאות ב `mini-batch`.
- **שימו לב:** הפונקציה צריכה להחזיר את הגרדיאנט, ולא את מינוס הגרדיאנט. כמו כן, אין צורך לחלק במספר הדוגמאות. הפונקציה `sgd` שמממשת את הלמידה עצמה (וכבר נתונה לכם) מבצעת את כלל העדכון הבא:

$$\text{weights}\{k\} = \text{weights}\{k\} - (\text{eta}/\text{mb_size}) * \text{grads}\{k\}$$

- מומלץ (מאוד) להעזר בסיכום תרגול 6 על `back-propagation` שבמודל.

אימון

השתמשו בקובץ המצורף `ex4_main` ע'מ לטעון את הדוגמאות ולאמן את הרשת. השתמשו בפרמטרים כפי שהם מופיעים בקובץ המצורף (10 מעברים מלאים על כל הדוגמאות (`epochs=10`), קצב לימוד $\eta = 0.1$, `batch_size=20`, ורשת עם שכבה אחת חבויה של 30 נוירונים)

1. הפונקציה `sgd` מחזירה שני וקטורים `[steps, test_accuracy]`. הציגו גרף של עקומת הלמידה (`test accuracy` כפונקציה של מספר העדכונים. שימו לב שהוקטור `steps` הוא ביחידות של `epochs`, כלומר התוצאה תהיה דיוק כפונקציה של כמה מעברים מלאים על כל דוגמאות האימון הרשת ביצעה).
2. בחרו באקראי 10 דוגמאות של כל אחת מהספרות 0, ..., 9, מתוך ה `test set`, וחשבו את הפרדיקציה של הרשת לגבי כל אחת. הציגו (ב-figure אחד) את כל 100 הדוגמאות שבחרתם (שורה ראשונה של ספרות 0, שורה שניה של ספרות 1 וכו'). סמנו באופן בולט דוגמאות שלגביהן הרשת טעתה (למשל, הוסיפו כותרת באדום עם התיג השגוי שהרשת סיפקה). מה ניתן לומר על סוג הטעויות שהרשת מבצעת?
3. חזרו על סעיפים 1 ו 2 כאשר מגרילים מראש פרמוטציה קבועה של הקלט, כלומר, כאשר משנים את סדר הרכיבים בתוך הקלט, באותו האופן על כל הקלטים (זכרו להפעיל את הפרמוטציה גם על הדוגמאות ב `train set` וגם על הדוגמאות ב `test set`. ראו הערה בקובץ `ex4_main`. השוו את התמונות שקיבלתם בסעיף 2 בכל אחד מהמקרים, ואת עקומות הלמידה. האם 'שיבוש' הסדר של התמונות השפיע על יכולת הרשת ללמוד? מדוע?

הערות למימוש:

- כדי להציג את התמונות, הפכו כל דוגמא חזרה למטריצה בגודל 28×28 והשתמשו בפונקציה `imshow`
- כדי לחשב את הפלט של הרשת, השתמשו בפונקציה (שנתונה לכם כבר) `predict`

¹נתונה גרסת מטלב וגרסת פייתון

- ע"מ לוודא שהאלגוריתם עובד, ולמצוא טעויות בקלות יותר, מומלץ להתחיל מלאמן את הרשת על חלק קטן ממדגם האימון ולוודא שהשגיאה על מדגם האימון יורדת, לפני שניגשים לאימון על מדגם האימון כולו ולבדיקה של שגיאת ההכללה.
- ע"מ לחשב את הדיוק (accuracy) של הרשת על קבוצת דוגמאות נתונה, תוכלו להשתמש בפונקציה (שנתונה לכם) `eval_test`
- החלקים שאתם צריכים למלא מסומנים גם בקוד ("YOUR CODE HERE").