דף עבודה - חישוב פונקציית XOR באמצעות רשת נוירונים רדודה

דף עבודה מוגש על ידי: בתאריך:

הערות כלליות:

1. את דף העבודה יש למלא בעזרת הרצה של הקוד הנמצא בקבצים הבאים:

XorUsesNN.py, NN.py

1. בכל מקרה בו אתם נדרשים לרשום ערכי משתנים, יש לרשום מה המשמעות של כל משתנה, ולמעט נגזרות יש גם לרשום את החישוב שהביא לערך זה ואת הערכים המספריים. ערכים מספריים יש למלא בדיוק של לא יותר מ- 3 ספרות אחרי הנקודה.
2. בכל מקרה בו אתם נדרשים לשנות קוד, שימרו את הקוד הקודם בהערה (#). בדרך זו תוכלו בקלות לחזור לקוד הקודם אם תדרשו לכך.

וכעת לדף העבודה עצמו:

1. כמו בכל בעייה בה נתקלנו עד עכשיו – אנחנו צריכים לבחור את המודל, לאפיין את הקלט והפלט, לאמן את המודל כלומר לחשב את ערכי W ואת ערכי b. לאחר מכן יש להשתמש במודל ובמשקלים שחישבנו על מנת לבצע חיזוי. במידה והחיזוי מצליח במידה טובה מספיק, נאמר שהמודל פותר את הבעייה.
2. מהם ערכי הקלט לבעיית XOR?

נזכור כי עבור 2 משתנים בוליאנים יש לנו 4 קלטים אפשריים כי כל אחד מהמשתנים יכול לקבל 2 ערכים, אמת או שקר (כלומר 0 או 1).  כל אחד מהקלטים מוצג כעמודה ויש לו את הפלט המתאים לו.

רישמו כאן את הערכים של עמודות הקלט השונות:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input #4 | Input #3 | Input #2 | Input #1 |  |
|  |  |  |  | x1 |
|  |  |  |  | x2 |

ואילו כאן רישמו את הפלט המתאים לכל אחת מהעמודות עבור פונקציית XOR:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Output #4 | Output #3 | Output #2 | Output #1 |  |
|  |  |  |  | Y |

1. בחירת המודל: ראינו שרגרסיה לוגיסטית לא מצליחה למצוא משקלים מתאימים עבור XOR – והבנו גם את הסיבה. רישמו כאן את הסיבה מדוע רגרסיה לוגיסטית, כלומר נוירון אחד, לא יכול לפתור את בעיית ה-XOR:
2. נתעכב מעט על אבחנה זו. להלן רשימת פונקציות בוליאניות של שני משתנים:

And; Or; Xor; Not Or; Not And; Not Xor

רישמו איזה פונקציות ניתנות לפתרון על ידי נוירון אחד ואיזה לא.

הפונקציות הניתנות לפיתרון על ידי נוירון אחד:

הפונקציות שאינן ניתנות לפיתרון על ידי נוירון אחד:

1. כיוןן שבעיית ה-XOR איננה ניתנת לפיתרון על ידי רגרסיה לוגיסטית, נאלץ להשתמש במודל מורכב יותר על מנת לפתור פורמאלית בעייה טריוויאלית זו. נכיר את המודל הבא – **רשת נוירונים**. רשת נוירונים היא הרכבה של נוירונים בשכבות. כל שכבה יכולה להכיל נוירון אחד או יותר, לפעמים מאות ואפילו אלפים של נוירונים בשכבה. כל נוירון מקבל קלטים מהשכבה הקודמת, מבצע חישוב של רגרסיה לוגיסטית ומעביר את התוצאה, הפלט, לשכבה הבאה. השכבה הראשונה, מקבלת את הקלט של הבעייה. השכבה האחרונה, שכבת הפלט, מחשבת את הערך החזוי.
2. בחרנו להתחיל בבעיה פשוטה יחסית אותה ניתן לפתור (ספוילר!) באמצעות רשת קטנטנה. רשת הנוירונים שלנו כפי שהוא מוגדרת בקוד המצורף, נראית כך:



X1

X2

prediction

אנחנו משתמשים בקונבנציה הבאה על מנת לסמן משקלים:

למשל בציור למעלה קיים בשכבה **השנייה** רק נוירון אחד, שני המשקלים שלו יסומנו:

ואילו עבור ערכי b נשתמש בסימון הבא:

ערכיz שיסומנו בדומה לערכי b על ידי השכבה (למעלה) והנוירון בשכבה (למטה) הם הצירוף הלינארי (נקרא גם קומבינציה לינארית) של הקלט שנכנס לנוירון. למשל:

=

שימו לב שגם כאן נשתמש ב-b לסימון המשקל של האיבר החופשי (אפשר גם להתייחס לקלט שלו כאל 1) כלומר הוספה של קבוע לצירוף הלינארי.

ערכי a שיסומנו בדומה לערכי z, b הם תוצאת פונקציית ההפעלה – במקרה שלנו סיגמואיד, על ערכי z:

רק בשכבה האחרונה "נתרגם" את ערכי a לחיזוי. כרגיל אם הם גדולים או שווים לחצי החיזוי יהיה 1 ואם הם קטנים מחצי החיזוי יהיה 0.

1. העזרו בדיאגרמה של הרשת בסעיף הקודם על מנת לענות על השאלות הבאות:

מה גודל הקלט בכל דוגמא (כלומר מה גובהה של עמודה אחת בקלט בתוכנה עצמה)?

כמה נוירונים בשכבה הסמויה, Hidden Layer?

כמה נויורונים בשכבת הפלט?

נחזור ונדגיש שההחלטות לגבי מבנה הרשת, כמות השכבות וגודלן, הן בשלב זה לפחות בחירה שרירותית שלנו. ככל שנעמיק ונצבור ניסיון נדע טוב יותר להתאים את הרשת לבעייה אותה אנו רוצים לפתור.

1. מלבד בניית הרשת, האתחול כולל גם קביעת משקלים, במקרה שלנו על ידי בחירת משקלים רנדומליים מתוך התפלגות אחידה. רשמו כאן את הערכים בהם בחרה התוכנה בבחירה הרנדומלית (לאחר קביעת seed) **עבור השכבה הראשונה:**

מהם ערכי b הראשונים של השכבה הראשונה?

ואילו עבור **השכבה השנייה,** כמה ערכי w יש לנו? כמה ערכי b? איך הם מסומנים ומה הערכים ההתחלתיים שלהם?

1. בדומה למודלים שראינו עד עכשיו בקורס, רשת הנוירונים משתמשת ב-Gradient Descent על מנת למצוא משקלים המביאים למינימום את פונקציית ה-cost בה בחרנו. במקרה של רשת אנחנו מבצעים את החישוב המלא של הרשת עובר כל קלט ("חלחול קדימה" Forward Propagations) ואז מתבצעת השוואה לפלט הנתון מראש וישנו "חלחול לאחור" Backward Propagation של הטעות) כלומר חישוב הנגזרות של המשקלים ועדכון של ערכי המשקלים בהתאם לערכי הנגזרות. בדיוק כמו שעשינו עד עכשיו גם במקרה זה נחזור על כל "סיבוב" כזה (חלחול קדימה, חלחול אחורה, עדכון משקלים) הרבה מאוד פעמים. כמה פעמים ומתי נעצור? אפשרות אחת היא כמות שרירותית וגדולה של פעמים. אפשרות טובה יותר היא בעזרת פונקציית cost. במקרה כזה נעצור כאשר השינוי בערך הפונקציה הזו בסיבוב נוסף הופך להיות זניח.

1. נרשום כאן את ערכי הסיבוב הראשון עבור הקלט 0, 0:

בשכבה הראשונה, עבור הקלט 0,0 מהם הערכים הבאים? יש לרשום את המשוואה בה השתמש הקוד ואת הערך שהתקבל בדיוק של 3 ספרות אחרי הנקודה:

ואילו עבור השכבה השנייה (יש למלא באותו האופן כמו בשכבה הראשונה):

1. כעת נעקוב אחרי החלחול לאחור. מה הערך שהרשת מצאה עבור הקלט 0,0 בסיבוב הראשון בו היא השתמשה במשקלים אקראיים? מה הערך הנכון לקלט זה?
2. אילו ערכים התקבלו עבור הקלט 0, 0 בחלחול לאחור הראשון עבור המשתנים הבאים? מה המשמעות של כל משתנה כזה?

1. מה ערכי הנגזרות שהתקבלו בסיבוב הראשון עבור הקלט 0,0 (בסעיף זה אין צורך לפרט את הנוסחאות. מספיק לרשום ערכים בלבד).

1. מהם המשקלים החדשים לאחר העדכון הראשון? העזרו בשאלה 8 בדף עבודה זה (משקלים במצב הפתיחה) כדי לדעת לאיזה משתנים הכוונה. רישמו את כל 9 המשקלים כאן (רמז קטן – הטבלה בסעיף הבא יכולה גם להיות שימושית...).
2. ראינו כיצד מתבצע סיבוב ראשון של עדכון. כעת נעבור לסיום התהליך. איזה משקלים מצא המודל בסיום פרוצדורת fit?

|  |  |  |  |
| --- | --- | --- | --- |
| Layer 1, Neuron 1 |  |  |  |
| Layer 1, Neuron 2 |  |  |  |
| Layer 2, Neuron 1 |  |  |  |

1. כעת נבדוק האם המודל (רשת נוירונים) והמשקלים שמצאנו אכן מחשבים נכון את פונקציית XOR. כיון שיש לנו רק 4 קלטים שונים, נזמן את פרוצדורת test עבור כל אחד מ-4 הקלטים השונים ונבחן את התוצאה שהמודל חישב לכל אחד מהקלטים. כלומר **בניגוד למה שבדרך כלל עשינו ונעשה** – לא נבחן את המודל על ערכי test חדשים, פשוט בגלל שבמקרה של XOR אין לנו עוד קלטים. רק במקרה זה ולמטרות לימוד אנחנו עושים מה שנקרא test on train (שזה בדרך כלל שגיאה חמורה). רישמו כאן את כל החישובים שהתבצעו עבור הקלט [0, 0] בפונקציית ה-test:
2. האם המודל שבנינו, רשת הנוירונים הרדודה, מצליח לחשב נכון את פונקצית XOR? כצד ניתן לקבוע זאת?

רבים מתייחסים לרשתות נוירונים כסוג של קופסא שחורה. זאת כיוון שבדרך כלל לא ניתן לאבחן מה בדיוק עושה כל נוירון ומה עושה כל שכבה. במקרה שלנו, כיוון שזו רשת מאוד מאוד מצומצמת וכיון שאנחנו עוסקים בערכים בוליאנים, ניתן לראות איזה פונקציה בוליאנית חישב כל נוירון.

הנוירון הראשון בשכבה הראשונה חישב את הפונקציה:

הנוירון השני בשכבה הראשונה חישב את הפונקציה:

מה לדעתכם גרם לשני הנוירונים לחשב פונקציות שונות?

הנוירון היחיד בשכבה השניה חישב את הפונקציה:

מלאו את טבלת האמת הבאה. יש למלא את הערכים וכן לרשום את הפונקציות הנכונות במקום הסימון ???

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Input #4 | Input #3 | Input #2 | Input #1 | Function |  |
|  |  |  |  | N.A. | x1 |
|  |  |  |  | N.A. | x2 |
|  |  |  |  | x1 ??? x2 | L1N1 output |
|  |  |  |  | x1 ??? x2 | L1N2 output |
|  |  |  |  | L1N1??? L1N2 | L2N1 output |

1. כעת נבצע מספר שינויים בקוד ונראה כיצד הם משפיעים:
   1. במקום 2 נוירונים בשכבה החבויה, נשתמש ב-3 נוירונים:
      1. רישמו כאן את שורות הקוד ששמתם בהערה ואת שורות הקוד החדשות שהוספתם במקומן:
      2. איזה פונקציה בוליאנית חישב הנוירון החדש שהוספתם?
      3. איזה פונקציה בוליאנית, הפעם של שלושה משתנים, חישב הנוירון היחיד שבשכבה השנייה?
      4. האם יש צורך בנוירון שלישי במקרה של חישוב פונקציית XOR? נמקו את תשובתכם.
   2. נחזיר את הקוד למצבו המקורי כלומר 2 נוירונים בשכבה החבויה. כעת נשנה את ערך ה-seed של הרנדום להיות 3 במקום 2.
      1. רישמו כאן את השינוי בקוד אותו ביצעתם.
      2. הריצו את הקוד לאחר השינוי ורשמו איזה פונקצייה חישב כל אחד מהנוירונים:

הנוירון הראשון בשכבה הראשונה חישב את הפונקציה:

הנוירון השני בשכבה הראשונה חישב את הפונקציה:

הנוירון היחיד בשכבה השניה חישב את הפונקציה:

* + 1. האם יש הבדל בין מה שחישב כל נוירון כשהקוד רץ עם seed=3 במקום seed=2?

מה גרם להבדל?

* 1. נחזיר שוב את הקוד לצורתו המקורית (seed=2 במקום seed=3). עכשיו נעשה שינויים בקובץ NN.py ונראה כיצד הם משפיעים.
     1. החליפו את ערכי האתחול של w באפסים. תוכלו להעזר בקוד שמאתחל את ערכי ה-b.

מהי שורת הקוד שהעברתם להערה?

איזה שורה הוספתם במקומה?

האם הרשת הצליחה לחשב את בעיית ה-XOR? מדוע?

* + 1. עכשיו החליפו את ערכי w ההתחלתיים בערך קבוע שאיננו אפס. לאחר יצירת w בין אם בעזרת הקוד המקורי או בעזרת הקוד שמאתחל את w באפסים, הוסיפו את שורת הקוד הבאה:

self.w.fill(3)

מה עושה שורת קוד זו?

האם הרשת המאותחלת במשקלים בעלי ערך קבוע מצליחה לחשב את בעיית ה-XOR? מדוע?

1. טריוויה (רשות):

19.1 הביטוי XOR הוא בעצם קיצור של ביטוי באנגלית. מה הביטוי המקורי? מה משמעותו?

19.2 מה הקשר המשפחתי בין ג'ורג' בול, המתמטיקאי הבריטי בן המאה ה-19 שניסח את חוקי האלגברה הבוליאנית הקרויה על שמו, ובין ג'ף הינטון, חוקר מדעי מחשב בריטי-קנדי בן זמננו שנחשב ל"סנדק" של למידת המכונה? (ג'ף הינטון הוא אחד משלושה חוקרים שניסחו לראשונה את המתמטיקה של רשתות הנוירונים).?

דף העבודה הוכן בחדווה על ידי עמיר שחל, עריכה אחרונה 20220406