עיבוד שפה טבעית

רטוב 1

ב. **הערות על שלב האימון:**

ללמידת וקטור המשקולות השתמשנו באלגוריתם האופטימיזציה optimize.fmin\_l\_bfgs\_b של חבילת scipy. על מנת למנוע התאמת יתר, הגדרנו פרמטרים ללמידה. הפרמטרים העיקריים ששימשו ללמידה היו הגבלת מספר האיטרציות והגדרת אחוז הדיוק לסיום איטרציה. את הפרמטרים הנ"ל קבענו על ידי ניסוי ותהייה. יש לציין כי בין שני הפרמטרים היה tradeoff, מאחר שדרישה לרמת דיוק פר איטרציה עלולה לגרום להתאמת יתר בעוד הגבלה על מספר האיטרציות מאזנת זאת. פרמטר נוסף שהשפיע על הלמידה הוא פרמטר למבדה. הפרמטר אחראי על הגבלת גודל הערכים בוקטור המאפיינים, ולכן הגדלתו תאלץ את הוקטור לערכים קטנים, אולם עלולה להשפיע על ההתכנסות. על מנת ללמוד את הפרמטר ביצענו מספר רב של הרצות ועל פי ערכי הגרדינט שהתקבלו מפונקצית האופטימיזציה כיווננו את הפרמטר (בשאיפה לקבל ערכים מינימילים ככל הניתן)

את אלגוריתם הלמידה הרצנו על מספר מחשבים שונים על מנת למקבל את ההרצות ולקבל מספר רב ככל הניתן של וקטורי מאפיינים עבור פרמטרים. המחשב ששימוש למרבית ההרצות הינו מחשב בעל מעבד i7 של אינטל.

על מנת למקבל השתמשנו במחשב בעל gpu GeForce 1080 אולם לא השתמשנו בו להסקה.

זמני הריצה עבור המודלים היו כדלקמן:

מודל בסיס – למידה כ15 דקות והסקה כ1 דקות

מודל מורכב – למידה כחצי שעה והסקה כ3 דקות

בכתיבת הקוד השתמשנו במספר חבילות של python:

Numpy – מספקת מערכים ופונקציות המשמשים לחישובים מתמטיים יעילים. בחבילה השתמשנו לאתחול וקטורים וביצוע פונקציות מתמטיות.

Scipy – לביצוע אלגוריתם הלימוד השתמשנו בoptimize של החבילה.

Pandas – לעבודה על מטריצת הבלבול, השתמשנו בחבילה המספקת ממשק נוח לעבודה עם טבלאות והפקת מידע.

על מנת להאיץ את הלמידה ממספר שעות (כ-3) החלטנו ראשית ליצור מילון של תיוגים שנראו למילים, ובעת אימון המודל לחשב סיגמואיד רק עבור תיוגים שכבר נראו (תוך חלוקה בכל התיוגים על מנת לשמור על מרחב הסתברות שלם). הדבר הביא לביצועים גבוהים (כעשירית זמן אימון) ודיוק גבוה יותר (כאחוז שיפור לכל מודל)

פיצ'רים במודל הבסיסי: (להוסיף מספר הפיצ'רים ואת הפיצ'רים במודל המורכב)

מודל מורכב:

כמו כן, בהתאם להנחיות הוספנו מספר פיצרים למודל המורחב:

ג. **הערות על שלב ההסקה:**

שינויים באלגוריתם ויטרבי:

1.כפי שראינו בתרגיל יבש1 , ניתן לייעל את ריצת ויטרבי ולקצר באופן משמעותי את זמן ריצת האלגוריתם אם נשתמש במילון T שמכיל מילים ותיוגים שנראו בזמן האימון. כאמור במקום לעבור על כל תיוג אפשרי בלולאות v,u,t נעבור על רק תגים שנראו בזמן האימון עם המילה הנוכחית. פתרון זה מועיל רק עבור מילים שנראו באימון ולכן עבור מילים שלא נראו באימון בחרנו לבצע איטרציה רק על מספר קבוע של תגים נפוצים (וכאמור בעזרת שיערוך המכפלה מהמודל הlog-linear נבחר את התיוג הטוב ביותר).

2. שיפור בסיבוכיות מקום: במקום לשמור את כל ערכי ה π(k,u,v) –האפשריים ( K ערכים, אחד לכל מילה במשפט), אנו שומרים רק π יחיד וכותבים עליו בכל איטרציה של k.

3. טיפול מיוחד במספרים: שמנו לב שהמודל שלנו טועה בתיוג עבור מספרים ולכן עבור מילים שאינן מופיעות במילון T[] ואכן זיהינו שמדובר במספר, אז רשימת התגים שאלגוריתם ויטרבי בוחן מוגבל לרשימת תגים של מספרים.

ד. **ניתוח תוצאות ע"פ שלב המבחן:**

Confusion matrix עבור המודל הבסיסי עם אחוז דיוק : 88.768%

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **True/Guess** | **IN** | **JJ** | **NN** | **NNP** | **NNPS** | **NNS** | **RB** | **VB** | **VBD** | **VBG** | **VBN** | **VBP** | **SUM** | **AVG** |
| **NN** | 41.00 | 81.00 |  | 374.00 |  | 36.00 | 7.00 | 56.00 | 1.00 | 20.00 | 1.00 | 3.00 | **629.00** | **23.66%** |
| **JJ** | 37.00 |  | 51.00 | 233.00 |  |  | 13.00 | 1.00 | 4.00 | 4.00 | 17.00 |  | **374.00** | **14.07%** |
| **NNS** | 35.00 |  | 9.00 | 170.00 | 3.00 |  |  |  |  |  |  |  | **237.00** | **8.91%** |
| **VB** | 17.00 | 6.00 | 82.00 | 39.00 |  |  | 1.00 |  | 1.00 |  | 2.00 | 84.00 | **235.00** | **8.84%** |
| **VBN** | 17.00 | 23.00 | 3.00 | 53.00 |  |  |  | 13.00 | 102.00 |  |  |  | **211.00** | **7.94%** |
| **VBD** | 7.00 | 2.00 | 4.00 | 39.00 |  |  |  | 7.00 |  |  | 104.00 |  | **163.00** | **6.13%** |
| **NNP** | 55.00 | 8.00 | 37.00 |  | 17.00 | 3.00 |  |  |  |  |  |  | **124.00** | **4.66%** |
| **RB** | 39.00 | 25.00 | 4.00 | 35.00 |  |  |  | 3.00 |  |  |  |  | **118.00** | **4.44%** |
| **VBG** | 8.00 | 2.00 | 27.00 | 68.00 |  |  |  |  |  |  |  |  | **105.00** | **3.95%** |
| **VBP** |  | 4.00 | 16.00 | 6.00 |  |  | 1.00 | 55.00 |  |  |  |  | **82.00** | **3.08%** |

ניתן לראות שהמודל הבסיסי מרבה לתייג מילים שהם JJ או NN בתור NNP. אנו משערים שאלו מילים שהוא לא ראה באימון ולכן תייג אותם כNNP כאשר לא הייתה אינדקציה משמעותית אחרת מתוך הוקטור.

Confusion Matrix for Complex model : אחוז דיוק: 89.059%

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **True/Guess** | **DT** | **IN** | **JJ** | **NN** | **NNP** | **NNPS** | **NNS** | **RB** | **VB** | **VBD** | **VBG** | **VBN** | **VBP** | **SUM** | **AVG** |
| **NN** |  | 24 | 71 |  | 267 |  | 36 | 8 | 55 | 1 | 18 | 2 | 2 | 493 | 19.03% |
| **JJ** |  | 26 |  | 122 | 174 |  |  | 14 |  | 4 | 4 | 18 |  | 376 | 14.52% |
| **VB** |  | 10 | 6 | 94 | 34 |  |  | 1 |  | 4 |  | 1 | 88 | 241 | 9.31% |
| **NNS** |  | 28 |  | 69 | 117 | 3 |  |  |  |  |  |  |  | 239 | 9.23% |
| **VBN** |  | 11 | 28 | 22 | 40 |  |  |  | 13 | 102 |  |  |  | 216 | 8.34% |
| **NNP** | 2 | 36 | 8 | 105 |  | 17 | 3 |  |  |  |  |  |  | 173 | 6.68% |
| **VBD** |  | 8 | 2 | 17 | 25 |  |  |  | 6 |  |  | 102 |  | 160 | 6.18% |
| **RB** | 5 | 37 | 24 | 11 | 29 |  |  |  | 3 |  |  |  |  | 116 | 4.48% |
| **VBG** |  |  | 2 | 110 | 1 |  |  |  |  |  |  |  |  | 113 | 4.36% |
| **VBP** |  | 1 | 4 | 18 | 3 |  |  | 1 | 55 |  |  |  |  | 82 | 3.17% |

ניתן לראות כי השיפור בין המודל הבסיסי למודל המורכב אינו גדול אך הוא ניכר בעיקר בתג NN. במודל הבסיסי 23% מהטעיות היו בתיוג שגוי של NN. ובמודל הבסיסי מספר הטעויות ירד ל19% וכמעט ב130 מילים פחות טעויות. לדעתנו השיפור הקל נובע מפיצ'רים עשירים יותר כגון תחיליות וסיומות, או הפיצ'ר שמתמודד עם capital letters. כמו כן התגים בהם קיימות טעויות נפוצות הם עדיין NN, JJ,VB בעוד שהתגים שהאלגוריתם מנחש וטועה הם לרוב NNP,NN. אנו משערים שאלו הם התגים הנפוצים בהם השתמשנו כאשר נתקלים במילה לא מוכרת.

נראה שניתן היה לשפר את המודל המורכב ע"י הוספת פיצ'רים שהיו מבדילים בין VBD לVBN או NNS ל NNP.

ה. **סיכום המשימה:**

ראשית ,ברצוננו לציין שמנקודת המבט שלנו , משימה זו ממחישה היטב את החזקות והחולשות של מודל מבוסס MEMM ופיצ'ר וקטור מקומיים. מצד אחד, בניית המודל אינה מורכבת במיוחד ורוב המשימות היו אינטואיטיביות, אך מנגד, תהליך האימון ערך זמן רב ונדרש אימון מחדש בכל שינוי של המודל כמו כן בחירת הפרמטרים הייתה לרוב מורכבת מניסוי וטעיה. אך בכל זאת אנו משערים שישנו יתרון מלהתנסות במימוש המודל הנ"ל בטרם מימוש אלגוריתמים שהם state of the art.

תרגיל זה עסק במשימת תיוג חלקי דיבר בהינתן קבצי אימון ומבחן כאשר ההסקה נעשית באמצעות אלגוריתם ויתרבי במימוש הMEMM שלו, אותו נתבקשנו לממש.

את המשימה חילקנו לשלבים הבאים:

1. תכנון מבני הנתונים הנדרשים ושיטת ייצוג יעילות עבור פיצ'ר וקטור
2. פירסור קבצי האימון וה-input.
3. למידת וקטור הפרמטרים v.
4. מימוש אלגוריתם ויטרבי ושימוש בו על מנת למצוא את התיוג המתאים ביותר לכל משפט.
5. כתיבת התוצאה לעיל במקום המתאים בקובץ המבחן.
6. ניתוח התוצאות, הוספות יוריסטיקות אשר נותנות מענה לטעויות נפוצות.

החלק שהיווה אתגר ודרש זמן רב היה למידת וקטור הפרמטרים. תחילה כל הרצה של למידת וקטור ערכה כשעתיים לפחות ולכן לקח זמן רב עד אשר מצאנו ערכים אשר התאימו. מאוחר יותר, החלטנו להגביל את כמות התגים שנמדדים בזמן הלמידה רק לתגים שנראו בזמן האימון עם המילה הנוכחית בתוספת K תגים נפוצים בניסיון לקצר את תהליך הלימוד. לאחר שינוי זה הלימוד ערך כ-15 דקות.

ו. **ניתוח משימת התחרות:**

במשימה זו לא ביצענו שינויים וכל האופטימזציות הופעלו כבר מראש במודל הבסיסי והמורכב. תיאור השינויים מופיע לעיל.

אנו צופים שאחוזי הדיוק יהיו קרובים לתוצאות שקיבלנו במודל המורכב על קובץ הtest. זאת מכיוון שמבדיקה שטחית נראה כי אחוז המילים החדשות (אשר לא הופיעו בקובץ האימון) דומה לאחוז המילים החדשות בקובץ הtest. כמו כן נראה שהData הוא פחות או יותר על אותו נושא כמו קובץ הtest. ובכל זאת אנחנו צופים שאחוזי הדיוק יהיו נמוכים יותר מאחוזי הדיוק של המודל המורכב. זאת מכיוון שאת ההתאמות והיורסטיקות שביצענו -היו בהתאם לטבלת הבלבול שמתאימה לקובץ הtest.

ז. **הסבר על חלוקת העבודה:**

תחילה התכנסנו והתחלנו לדון ולתכנן מה הם החלקים בעבודה, וחילקנו אותה לתתי משימות. לאחר מכן נפגשנו כמה פעמים על מנת לדון במימוש ותכנון מבני הנתונים הנדרשים על מנת לתחזק ולפענח את המידע הנדרש כמו כן צורת המימוש של הפיצ'ר וקטור. החלטנו לחלק את העבודה בינינו כך שימין יהיה אחראי על פירסור המידע ובניית מבני הנתונים ותהליך הלמידה ורן יתחיל במימוש ויטרבי. לאחר מכן כשחלקים אלו נכתבו, בדקנו אותם על כמה משפטים בודדים מתוך קובץ האימון. כל הפגישות לאחר מכן התמקדו בשיפור ביצועים, הוספת פיצ'רים למודלים ומימוש יורסטיקות במטרה לשפר את אחוזי הדיוק על בסיס טבלת הבלבול בנוסף לניסוי ותהיה בנוגע לפרמטרים כמו הקבועים שמשמשים ללמידה והסקה. לאחר כל שינויים אלו נדרשנו לאמן שוב את פרמטר הלמידה V, אך לאחר אופטימזציות אלו תהליך הלמידה היה מהיר בכמה מאות אחוזים.