הפקולטה למדעים מדויקים מועד א'

החוג לסטטיסטיקה וחקר ביצועים סמסטר א', תשפ"ב

אוניברסיטת תל אביב 25.1.2022

**למידה עמוקה**

מרצה: דר' שמעון שחר

כל חומר עזר דומם מותר בשימוש, אסור להתייעץ עם אנשים אחרים.

את הפתרונות יש לדחוס לקובץ zip יחיד, ולשלוח במייל אל monishahar@tauex.tau.ac.il.

שורת הכותרת של המייל צריכה להיות: Deep Learning Course Test <id number>.

לדוגמה: Deep Learning Course Test 024484214

שאלה 1 (50)

כאשר כותבים chatbot אחת המטלות העיקריות נקראת intent classification. במטלה זאת יש לבנות מודל שיקבל משפט או פסקה קצרה ויחזיר אחת מ-k אפשרויות מוגדרות מראש. למשל במערכת מענה קולי של רופא, האפשרויות אותן יש לזהות הן: "הזמנת תור", "קבלת אישור" או "אחר". אם הקלט הוא: "אני רוצה לקבוע תור לרופא ביום ראשון" ה- class שהמודל יחזיר הוא "הזמנת תור".

בלינק ל- Google Drive שמצא כאן, נתון dataset לאימון ובחינה של מודל ל- intent classification.

**שימו לב:** הקבצים בסיומת out שהועלו לדרייב אינם רלוונטיים לבעיה. בדרך כלל פותרים בנוסף ל- intent classification בעיה נוספת שקשורה להבנת המשפטים, וקבצים אליה רלוונטיים אליה.

<https://drive.google.com/drive/folders/1vtjtClS22qSQVuhSd4VoAb9_xfRPXs9x?usp=sharing>

1. כתוב מודל ל- intent classification אשר אינו משתמש ב- language model

נכתב מודל LSTM – בקובץ המצורף. הערה: מספר ה-epochs נקבע ל-30 מכיוון שהאימון של הembeddings נעשה במודל, לוקח למודל זמן להתעדכן ולכן רק לאחר 17 epochs מתחיל שיפור משמעותי.

1. כתוב מודל ל- intent classification אשר משתמש ב language model. מומלץ להשתמש ב- BERT אך לא חובה.

נכתב מודל BERT. ניתן להריץ על GPU או על CPU – עדיף על GPU. מספר ה-epochs נקבע על 5.

בקובץ הזיפ המצורף יש שלושה קבצים בנוסף לטופס:

1. Exam\_203868187\_1a.py – קוד ל-LSTM.
2. Exam\_203868187\_1b.py – קוד ל-BERT.
3. Reqierments.txt - קובץ טקסט המרכז פירוט של החבילות והערות על כל אחד מן הסעיפים.

הקוד לשני המודלים שכתבתם צריך להריץ אימון, פרדיקציה, ולהדפיס את תוצאות ה- evaluation על ה- test data שנתון. במידה ואתם משתמשים בספריות (למשל ספריית transformers שהיא ספרייה מקובלת ל- language models) ציינו את זה, או כתבו קובץ requirements.txt.

שאלה 2 (30)

Conditional Random Field הוא מודל גרפי שנמצא בשימוש בבעיות machine vision. המודל מוגדר על ידי גרף שבו כל קודקוד מייצג משתנה מקרי Y שמאונדקס לפי אינדקס הקודקוד v, ואוסף תצפיות X. התכונה העיקרית של המודל היא שמתקיימת מרקוביות בשכנים שבגרף. כלומר:

P[Yv |X, Y\{Yv}] = P[Yv | X, {Yw; w neighbor of v}]. הגדרה מפורטת של המודל נמצאת למשל בוויקיפדיה: <https://en.wikipedia.org/wiki/Conditional_random_field>

בעיית ה- semantic segmentation מוגרת כבעיה שהקלט אליה הוא תמונה והפלט הוא label לכל פיקסל, כאשר ה- labels נתונים מתוך קבוצה דיסקרטית ידועה מראש (לדוגמה: אנשים, עצים ורקע).

1. הסבר עבור בעיית semantic segmentation מה מייצג כל משתנה Y? מה מייצגות התצפיות X?

המשתנה Y מייצג את המחלקות השונות אליהן ניתן לסווג את הקודקודים. במשוואה המוצגת מעלה – מחשבים את ההסתברות המותנית של Y בקודקוד ספציפי – בהינתן המשתנה X ו-Y של השכנים של הקודקוד. צריך לשים לב כי ההסתברויות המותנות מחושבות לכל התיוגים האפשריים של Y.

התצפיות X מייצגות את הפיצ'רים של כל קודקוד (המאפיינים של הקודקוד שבעזרתם נסווג אותו למחלקה ב-Y) – במשוואה המוצגת מעלה, ההסתברות המותנית מחושבת בעזרת הפיצ'רים הללו. במקרה של תמונה – הפיצ'רים יכולים להיות ערך ה-RGB, מיקום, texture, וכו'.

1. קרא את המאמר: Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials והסבר את המודל להסתברות המותנה ואת הקירוב שמשתמשים בו ללמידה (sections 2,3).

במאמר, המודל שתואר הינו Fully Connected CRF, כלומר, הוא מייצג כל פיקסל על ידי קודקוד בגרף. כל קודקוד, מחובר לכלל הקודקודים האחרים (לכל הפיקסלים האחרים), זאת על מנת לבטא קשרים יותר "ארוכי טווח" מאשר תלות אך ורק בפיצ'רים של הפיקסל הספציפי או רק בשכנים שלו.

המודל להסתברות המותנה נעזר בהתפלגות גיבס: מחושבת ההסתברות של כל פיקסל לקבל תיוג מסוים על פי "האנרגיה" שלו (מתבצע גם נרמול על פני כלל התיוגים).

"האנרגיה" של המערכת מבוטאת על ידי סכום של הפוטנציאלים היחידים – כלומר, ההסתברות של כל פיקסל להיות שייך למחלקה מסוימת על בסיס הפיצ'רים שלו בלבד (יכול להתבצע על ידי SVM למשל), והפוטנציאל "הזוגי" (pairwise).

הפוטנציאל הזוגי מחושב על ידי כפל של פונקציית התאמת תגיות בקומבינציה לינארית של משקולות (( וקרנל גאוסיאני שלוקח בחשבון את הפיצ'רים של שני פיקסלים שונים.

הקירוב שבוצע ללמידה הוא קירוב להסתברות האמיתית של פיקסל מסוים להיות שייך למחלקה מסוימת. הקירב מנסה למעשה למזער את ה KL- divergence.

ההתפלגות המקורבת מחושבת על ידי הכפלה של קירובים.

כל קירוב יחיד הוא אקספוננט של פוטנציאל יחיד וסכום של פונקציית התאמת תגיות וקומבינציה לינארית של משקולות, אשר כופלים בקרנל גאוסיאני.

1. קרא את המאמר: Conditional Random Fields as Recurrent Neural Networksוהסבר כיצד מקרבים את הבעיה כרשת נוירונים. בפרט: (1) תאר את ההבדל בין שכבת קונבולוציה רגילה לשכבה המתוארת במאמר ו- (2) תאר כיצד מתבצעת הגזירה (sections 3-5).

בניגוד לשכבת קונבולוציה רגילה, בה הפילטרים והפרמטרים הם קבועים לאחר שלב האימון, בשכבות הקונבולוציה המתוארות במאמר, הפילטרים הם פונקציה של המיקום ושל הפיצ'רים של הפיקסל (במאמר, הדוגמה היא לערך ה-RGB של הפילטר). כל פילטר כזה הוא בגודל התמונה כולה, בגלל ש-CRF הוא fully connected (כלומר, כל פיקסל תלוי בכל הפיקסלים האחרים).

בנוסף, הם משתמשים ב edge preserving gaussian filters – הפילטרים האלו שומרים מידע של פיקסלים "תוחמים" – כלומר, מדגישים קווי מתאר ומטשטשים פיקסלים אחרים.

במאמר, החוקרים קירבו את הצעדים באלגוריתם המוצג כרשתות קונבולוציה.

בשכבת האתחול , הם פשוט ביצעו את הקירוב בדומה ל – softmax ברשתות קונבולוציה.

בשכבת ה-message passing - הם מימשו על ידי החלת M פילטרים גאוסיאניים על Q ערכים. הפילטרים נקבעו על ידי הפיצ'רים של הפילטר, שמראים כמה פיקסל "קרוב" לשכנים (מכיוון שכל הפיקסלים שכנים אחד של השני, השתמשו בקירובים לפילטרים הגאוסיאניים על מנת לייעל את זמני הריצה). הגזירה בשלב זה נעשית באופן הפוך: הנגזרות של השגיאות לפי הקלט מחושבות על פי השגיאות של הפלט. הנגזרות של השגיאות על הפלט מועברות דרך הפילטרים הגאוסיאניים בכיוון ההפוך (מופעלת עליהם הפונקציה ההפוכה לפילטר).

בשכבת Weighting Filter Outputs – נלקחו הפלטים של M הפילטרים עבור כל לייבל מהשלב הקודם. מכיוון שכל פיקסל נבחן על כל לייבל אז למעשה ניתן למדל זאת כרשת קונבולוציה כאשר הפילטר הוא 1X1, וקלט הוא M ערוצים והפלט הוא ערוץ יחיד.

ניתן לחשב את הגזירה של הטעויות לפי הקלט בצורה ישירה וכך להעביר לשכבה הקודמת שממנה נלקח הקלט.

שכבת Compatibility Transform - בשלב זה, יש פונקציה של תאימות בין כל זוג לייבלים.

מכיוון שמחשבים את הפרמטר עבור כל הזוגות, גם פה ניתן למדל כשכבת קונבולוציה עם פילטר של 1X1, כאשר הקלט והפלט הם בגודל של מספר הלייבלים. המשקולות פה הן פשוט פונקציית התאימות (מסומנת כ- במאמר).

שלב Adding Unary Potentials – בשלב זה, מחסירים את הפלט של השלב הקודם מהפלטים היחידים (unary). מכיוון שהחיסור קורה element-wise, ניתן לבצע את השלב הזה בעזרת קונבולוציה.

של הנרמול – בדומה לשלב האתחול, מתבצע קירוב של softmax בעזרת רשת קונבולוציה (במאמר אין פירוט כיצד זה נעשה – אלא נאמר שזו בעיה מוכרת ומקורבת רבות בעזרת רשתות קונבולוציה).

כל שלב יושם באמצעות רשת אחרת. מכיוון שבכל שלב מתקבל קלט גם מהשלב הקודם, ניתן למדל את כל האלגוריתם כרשתות קונבולוציה שמקבלות קלט מהשלב הקודם, וניתן גם לבצע איטרציות רבות של האלגוריתם, שכל אחת מהן לוקחת ערכים מהאיטרציה הקודמת – כלומר, ניתן להסתכל על כל האיטרציות כ-RNN.

שאלה 3 (20)

בשאלה זאת מספר סעיפים אשר אינם קשורים ביניהם.

1. הסבר כיצד ניתן להפוך autoencoder נתון ל- variational autoencoder? מהם השינויים בארכיטקטורה ובפונקציית המטרה. הנח שהרשת המקורית מיושמת באמצעות מספר שכבות fully connected.

על מנת להפוך autoencoder ל – variational autoencoder, צריך כמה שינויים בארכיטקטורה:

ראשית, במקום ייצוג ווקטורי יחיד לכל דגימה, אותו למדנו על ידי autoencoder, ב-vae, נלמד שני וקטורים: הראשון ייצג ממוצע, השני ייצג סטיית תקן (שניהם רב ממדיים).

בעזרת הווקטורים האלו, נוכל לייצר דגימות חדשות.

נצטרך גם להוסיף דגימה אקראית, מהתפלגות נורמלית סטנדרטית רב ממדית ().

לאחר שלמדנו ממוצע וסטיית תקן, נייצר דגימה, המתפלגת לפי הערכים שנלמדו (ממוצע וסטיית תקן) על ידי ה-vae. כדי ליצור אקראיות, על מנת לייצר מרחב של נקודות שלא ראינו, נכפיל את סטיית התקן בדגימה האקראית שדגמנו - -(נקרא reparameterization trick) – את הדגימה הזו נכניס ל- decoder כדי לקבל construction error – בדומה ל-autoencoder.

פונקציית המטרה תהיה למזער את ה-loss: ה-loss של ה-vae מורכב משני חלקים-

בדומה לautoencoder, גם ל-vae יש מרכיב של reconstruction error – כלומר, שגיאת בנייה שבודקת את איכות ה-decoder.

בנוסף למרכיב זה, אנחנו מוסיפים מרכיב של רגולריזציה – שהוא למעשה KL- divergence בין ההתפלגות שדגמנו להתפלגות נורמלית סטנדרטית. כך, אנחנו שומרים על הרציפות והשלמות של המרחב ולא מאפשרים לדגום מהתפלגויות שלא ייתנו לנו דגימות הגיוניות אלא רק רעש.

1. האם אפשרי להשתמש בשכבות convolution ב- autoencoder, ואם כן מה נרצה להניח שהדאטה בקירוב מקיים על מנת שיהיה כדאי להשתמש בהן?

אפשר להשתמש בשכבת קונבולוציה ב-autoencoder.

שכבות קונבולוציה יכולות להוריד ממד, ובפרט, הן יכולות להוריד את הממד של הקלט לממד של הייצוג ב-autoencoder. על מנת להגדיל חזרה את הייצוג ב-autoencoder לגודל של הקלט, ניתן פשוט לעשות דה-קונבולוציה.

על מנת להשתמש בשכבת קונבולוציה להורדת ממד, אנחנו צריכים להניח שיש משמעות מרחבית לקלט, מכיוון שהפילטרים של שכבות קונבולוציה עובדים הכי טוב כאשר לקלט יש משמעות מרחבית – כמו בתמונה.

1. אותה שאלה כמו בסעיף ב' עבור LSTM.

ניתן להשתמש בשכבות קונבולוציה עבור LSTM.

LSTM מקבל כקלט ייצוג וקטורי. שכבות הקונבולוציה יכולות להפוך קלט לייצוג וקטורי שייכנס כקלט ל-LSTM.

בדומה לסעיף ב', גם פה צריך להניח כי לקלט יש משמעות מרחבית. מכיוון ש-LSTM נועד לפתור בעיות של רצף (sequence), אז רצוי שבקלט תהיה גם משמעות לסדר – למשל, סרטוני וידאו (שלהם יש גם משמעות למרחב וגם משמעות לרצף).

1. נתון dataset שבו התצפיות מגיעות משתי התפלגויות רב-מימדיות שונות. כמו-כן נתון כי רק 0.001 מתוך התצפיות באות מההתפלגות השניה, אך לא ידוע אילו תצפיות מקורן בכל אחת מההתפלגויות. הצע דרך שתשתמש ב- autoencoders כדי לזהות את המקור לתצפיות והסבר מדוע אתה חושב שהיא תעבוד.

**רמז:** כדוגמה קונקרטית הנח שכל אחת מההתפלגויות מרוכזת מאוד סביב יריעה שונה במימד 2 והדאטה הנתון הוא במימד גבוה בהרבה. מה יהיה המימד הרצוי של ה- encoding?

ניתן לבצע תהליך דומה ל-PCA. מכיוון שהיריעות מאד שונות במימד 2, אז נבנה autoencoder שיוריד את הדאטה למימד 2.

לאחר מכן ניתן להכניס את הייצוג במימד 2 למסווג פשוט – למשל ,SVM שיסווג את הייצוג לכל אחת מן ההתפלגויות.

לדעתי דרך זו תעבוד מכיוון שאם יש הבדל משמעותי בין ההתפלגויות במימד 2, אז לאחר שהצלחנו להביא את הייצוג של הדאטה של מימד 2, מספיק מסווג לינארי פשוט שיידע להפריד בין ההתפלגויות.

העובדה שהתפלגות אחת נדירה הרבה יותר מהשנייה אינה משנה – מכיוון שניתן למקם את מישור (במקרה הזה – קו) ההפרדה במקום שיידע לסווג את רוב התצפיות להתפלגות הראשונה.