תרגול שלוש עשרה - תרגול חזרה

בתרגול נכיר שאלה ברוח המבחן.

שאלה 1 - רשתות נוירונים

א. עבור בעיית סיווג עם יותר משתי מחלקות מגדירים את פונקציית הלוסס הבאה:

$$L(y,\hat{y}) = \sum_i -y_i log \frac{e^{\hat{y}_i}}{\sum_j e^{\hat{y}_j}}$$

כאשר y הוא ווקטור הלייבלים מקודד בצורת one hot encoding, וy כובע הינו ווקטור הפרדיקציות המכיל ערכים ממשיים. מה הייתרון של פונקצית השגיאה הזו על פני פונקצית שגיאה מסוג MSE עבור הבעיה המדוברת?

ב. ברצוננו לתכנן רשת לסיווג אובייקטים בתמונות שחור-לבן תלת מימדיות מגודל 10x10x10 פיקסלים. אנחנו יודעים שאותו טיפוס של אובייקט יכול להופיע במקומות שונים בתמונות שונות ויכול גם להופיע בגדלים שונים. תכננו רשת נויירונים עבור בעיה זו. הסבירו את הבחירות שלכם

ג. אימנתם את הרשת מסעיף ב' על אוסף גדול של תמונות 10x10x10 שנמצא ברשותכם וקיבלתם מסווג עם שגיאה נמוכה. מבקשים מכם להתאים את הרשת כך שתוכל לעבוד על תמונות מגודל 11x11x11 אבל יש רק אוסף קטן של תמונות לאימון מגודל זה. הציעו לפחות 3 דרכים שונות איך ניתן להתמודד עם אתגר זה והסבירו את היתרונות והחסרונות של השיטות השונות.

שאלה 1 - פתרון

א. עבור בעיית סיווג עם יותר משתי מחלקות מגדירים את פונקציית הלוסס הבאה:

$$L(y,\hat{y}) = \sum_i -y_i log \frac{e^{\hat{y}_i}}{\sum_j e^{\hat{y}_j}}$$

כאשר y הוא ווקטור הלייבלים מקודד בצורת one hot encoding, וy כובע הינו ווקטור הפרדיקציות המכיל ערכים ממשיים. מה הייתרון של פונקצית השגיאה הזו על פני פונקצית שגיאה מסוג MSE עבור הבעיה המדוברת?

ראשית נבחן את הפונקציה המדוברת. עבור ווקטור y כובע, יוצרים ווקטור חדש בו כל מספר הוא אקספוננט של הערך הווקטור y כובע באותו מיקום. לבסוף מחלקים בסכום כל האקספוננטים.

המשמעות - כל איבר בווקטור הוא מספר בין 0 לאחת. אנחנו נפרש את המספר הזה כהסתברות:

$$\hat{y} = [1, 0.5, -300, 10]$$

$$e^{\hat{y}} = [2.72, 1.65, 0, 22026]$$

$$\sum e^{\hat{y}} = 22030.37$$

$$\frac{e^{\hat{y}}}{\sum e^{\hat{y}}} = [1.24 \cdot 10^{-4}, 7.45 \cdot 10^{-5}, 0, 0.999]$$

כעת נבדוק מה יהיה הווקטור y.
לפי ההוראות הוא ווקטור המציין מחלקות מסוג one hot encoding. ניצור מלאכותית שניים כאלה:

$$y_1 = [0, 0, 0, 1]$$
$$y_2 = [1, 0, 0, 0]$$

כעת נחשב מה יהיה הלוסס בשאלה עבור כל ווקטור:

$$L_1 = -(0 \cdot log(1.24 \cdot 10^{-4}) + 0 \cdot log(7.45 \cdot 10^{-5}) + 0 \cdot log(0) + 1 \cdot log(0.999))$$
$$L_1 \approx 4.34 \cdot 10^{-4}$$

$$L_2 = -(1 \cdot log(1.24 \cdot 10^{-4}) + 0 \cdot log(7.45 \cdot 10^{-5}) + 0 \cdot log(0) + 0 \cdot log(0.999))$$
$$L_2 \approx 3.92$$

כעת נשווה מה יהיה הלוסס אם היינו משתמשים ב MSE:

$$L_1 = ((0 - 1.24 \cdot 10^{-4})^2 + (0 - 7.45 \cdot 10^{-5})^2 + (0 - 0)^2 + (1 - 0.999)^2) \cdot \frac{1}{4}$$
$$L_1 \approx 2.55 \cdot 10^{-7}$$

$$L_2 = ((1 - 1.24 \cdot 10^{-4})^2 + (0 - 7.45 \cdot 10^{-5})^2 + (0 - 0)^2 + (0 - 0.999)^2) \cdot \frac{1}{4}$$
$$L_2 \approx 0.49$$

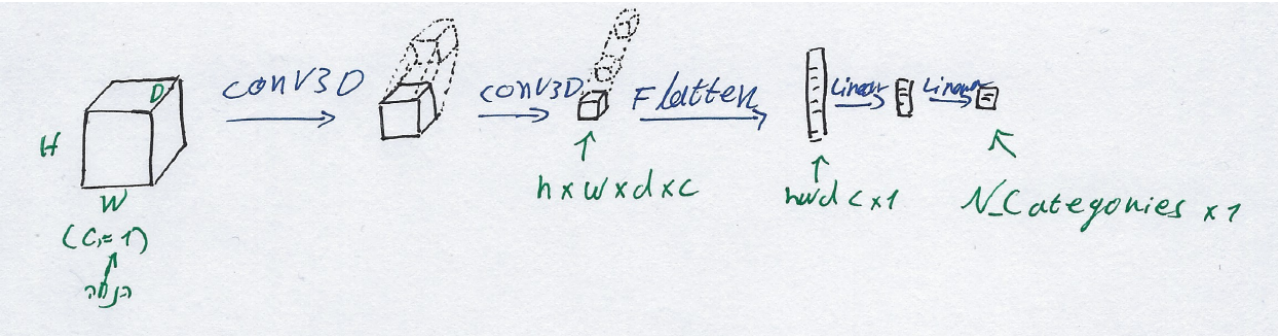
מכאן קיבלנו הבדלים ברורים:
1. בשני המקרים נענשנו כשצדקנו במחלקה. עם זאת במקרה הראון, עם הלוסס בשאלה, מקור העונש הוא מה שקיבלנו במחלקה הנכונה - הרשת תידחף להעלות את הפרדיקציה עבור המחלקה הזו. עבור המקרה השני, עם השגיאה הריבועית, מקור העונש הוא במחלקות האחרות (הלא נכונות). במקרה זה הרשת תידחף לדיכוי אותן מחלקות. נשים לב לגודל העונש במקרה הזה - קטן מאוד עד כדי לא משמעותי.
2. במקרה בו טעינו (ווקטור y2) העונש עבור הלוסס בשאלה קטן מאוד לעומת העונש עבור שגיאה ריבועית - מה שיגרום לתהליך למידה איטי יותר.
3. באופן תאורטי, שגיאה ריבועים ממוצעת חסומה על ידי 1 ואילו הלוסס המדובר אינו חסום - אם המחלקה הנכונה קיבלה הסתברות אפס היינו מקבלים שגיאה אינסופית (במציאות שמים הגנות על מנת להימנע ממצב של אינסוף בשגיאה) - הרשת הייתה נענשת בחומרה.
מכל מה שאמרנו למעלה - תהליך ההתכנסות בשגיאה הנתונה יהיה יותר מהיר וישים יותר דגש על המחלקה הנכונה (לעומת דיכוי מחלקות אחרות) בכל סבב - דבר שיקל על הלמידה.

ב. ברצוננו לתכנן רשת לסיווג אובייקטים בתמונות שחור-לבן תלת מימדיות מגודל 10x10x10 פיקסלים. אנחנו יודעים שאותו טיפוס של אובייקט יכול להופיע במקומות שונים בתמונות שונות ויכול גם להופיע בגדלים שונים. תכננו רשת נויירונים עבור בעיה זו. הסבירו את הבחירות שלכם

כשלמדנו על אודות רשת נוירונים למדנו שני מודלים עיקריים - מודל מסוג MLP ומודל קונבולוציה. כעת עלינו לתכנן רשת כך שבאופן פרקטי נקבל מודל לומד לפי הדרישות.
נתייחס ראשית למודל - לבחירתינו שני מודלים. מהאמור בשאלה אנחנו מבינים שהמודל בו נבחר צריך להיות אדיש למיקום האובייקט בשאלה (וגם לגודלו). מבין שני המודלים שלמדנו רק אחד מהם היה אדיש למיקום וזהו מודל קונבולוציה.
תוצאת הפעלת קרנל מסביב לפיקסל כלשהו ועל השכונה שלו תהיה זהה בין אם הפיקסל והשכונה יהיו במרכז התמונה או באחת הפינות - התוצאה שתתקבל היא אותה תוצאה. מכאן אנחנו מבינים עיקרון חשוב ברשתות נוירונים - רשת קונבולוציה היא position invariant, לעומת רשת מסוג MLP שם המשקולות שנלמדו מקובעים למקום ואם האובייקט יזוז בתמונה יופעלו עליו משקולות אחרות.

כעת נבחן את הארכיטקטורה - מבנה השכבות השונות. אנחנו מבינים שאנחנו צריכים שכבות קונבולוציה לאורך הרשת וגם שאנחנו בבעית קלסיפיקציה, לכן נצטרך שכבות ליניאריות בהמשך הרשת. נדון באילוצים:
אנחנו מאולצים להשתמש בשכבות קונבולוציה לאורך הרשת.
קרנל קונבולוציה יהיה תלת מימדי פלוס מימד אחד של ערוצים (סך הכל ארבעה מימדים).
מוצא הרשת יהיה ווקטור באורך מספר המחלקות.
יחד עם זאת יש לנו דרגות חופש:
אנחנו רשאים לבחור קרנלים בגדלים משתנים (בין אחת על אחת על למספר גבוה בהתאם לפרמטרים השונים). הפרמטרים של הקרנלים לבחירתינו, אבל אנחנו מצפים שמפות האקטיבציה יקטנו במימדים HWD ככל שמתקדמים ברשת.
אנחנו רשאים לבחור כמה קרנלים להחיל בכל שכבה, אבל יחד עם זאת אנחנו מצפים שהמספר הזה יגדל ככל שמתקדמים ברשת.
אנחנו רשאים לבחור את החלק ברשת המבצע קלסיפיקציה. נבחר אותו בתור רשת MLP עם מספר הולך וקטן של נוירונים עד למוצא.

נייצר רשת כזו ונשרטט אותה:



כאמור, הרשת הינה תלת מימדית והאינפוט מיוצג בתור טנזור מספרים HxWxD. כאן אנחנו מניחים כי כל מימד הוא בגודל 10 ומשום שהתמונות שחור-לבן נסיק כי יש ערוץ אחד באינפוט. הרשת מבצעת קונבולוציות בתלת מימד ותוך כדי מגדילה את מספר הערוצים לקבלת טנזור בגודל hxwxdxc כאשר הערך c נתון לבחירתנו. לבסוף "משטחים" את הטנזור לווקטור ומחילים עליו רשת קלסיפיקציה לקבלת ווקטור באורך מספר המחלקות.

ג. אימנתם את הרשת מסעיף ב' על אוסף גדול של תמונות 10x10x10 שנמצא ברשותכם וקיבלתם מסווג עם שגיאה נמוכה. מבקשים מכם להתאים את הרשת כך שתוכל לעבוד על תמונות מגודל 11x11x11 אבל יש רק אוסף קטן של תמונות לאימון מגודל זה. הציעו לפחות 3 דרכים שונות איך ניתן להתמודד עם אתגר זה והסבירו את היתרונות והחסרונות של השיטות השונות.

(i) ניתן לאמן רק על תמונות מהגודל הנכון אבל אז נקבל מסווג לא טוב בגלל שיש רק מעט תמונות לאימון

(ii) ניתן לאמן מחדש רק את שכבות ה-dense של הרשת הקודמת – זה קצת יותר טוב מהפתרון הראשון אבל עדיין בשכבות האלה יש הרבה מאוד פרמטרים לאמן עם מספר קטן של דוגמאות

(iii) ניתן להפעיל את הרשת מהסעיף הקודם בהזזות של 1 על התמונה החדשה ולאמן רשת קטנה שתעשה סוג של pooling על התוצאות של 8 ההפעלות

(iv) ניתן לעשות סקיילינג לתמונות – זה יכול להכניס רעש

(v) ניתן לאמן רשת חדשה כאשר משתמשים גם בתמונות הקטנות יותר לאימון על-יד זה שעושים להם padding – זה יכול לתת תוצאות טובות אבל צריך לבדוק שהרשת לא נסמכת יותר מדי על ה-padding

(vi) הוספת שכבה של pooling אבל כזו שתפעל באופן גלובלי על כל המימדים המרחביים ותהפוך אותם למספר אחד ונישאר עם ווקטור באורך ידוע של מימד הערוצים, אבל פתרון זה יהיה יעיל רק בתנאי שההבדל בגודל בין התמונות ובין האובייקטים לא גדול מידי.