המחלקה להנדסת תוכנה

פרויקט גמר –

בחירת דאטה משמעותי לאימון

Selection of significant data for training

מאת:

בצלאל כהן

ת.ז 308571207

מנחה אקדמי: דר' יהודה חסין אישור: תאריך: 22.1.2023

מערכות ניהול הפרויקט:

#	מערכת	מיקום
1	מאגר קוד	https://github.com/bezalelc/Data-pruning
2	קישור ליומן	https://docs.google.com/document/d/1flicN25UgrOSoRnzft5hCvXEyHCTuxpd2Q8Dh7Sxfi8/edit?usp=sharing
3	קישור לסרטון דוח אלפא	https://drive.google.com/file/d/1FgsILAiTd-goguBsqFPmWN_Pv2FbZsSl/view?usp=sharing

מידע נוסף:

סוג הפרויקט	מחקרי ממרצה במכללה
פרויקט מח"ר	לא
פרויקט ממשיך	זה פרויקט חדש
פרויקט זוגי	לא

מילון מונחים, סימנים וקיצורים

EL2N - Ensemble L2 Norm
STD - Standard deviation

תוכן עניינים

0. תקציר

הפרויקט הוא פרויקט מחקרי עם דר' יהודה חסין, מטרת הפרויקט היא לבדוק האם ואיך ניתן לקצץ חלק מדאטה קיים ולקבל תוצאות ברות השוואה על מודלים ללמידה עמוקה בשביל לחסוך במשאבים (זמן, כוח חישוב, איסוף דאטה) ו/או להגיע לביצועים יותר טובים של שגיאת המודל.
יש מחקרים שנעשו כבר על הנושא עם הצעות ומדדים שונים איך למצוא את הדאטה המשמעותי יותר לאימון ולקצץ את השאר, במסגרת הפרויקט נממש אותם, נציע מדדים נוספים ונבדוק כמה כל שיטה משפיעה על תהליך הלמידה של מודלים ללמידה עמוקה.

1. מבוא

למידת מכונה היא תחום במדעי המחשב שמאפשר לפתור בעיות שלא ניתן לפתור בעזרת תכנות "רגיל" כלומר בעזרת לולאות for ותנאים לוגיים של if-else, לדוגמא חיזוי מזג אוויר לפי נתוני שנים קודמות או סיווג אובייקטים בתמונה, בלמידת מכונה מלמדים את המחשב לעשות את מה שבא לבני אדם באופן טבעי - ללמוד מניסיון.
כדי "ללמד" את המחשב בונים מודל חישובי ומאמנים אותו על מידע קיים כדי שילמד תבניות מסוימות ואז המודל יכול לתת לנו הערכה על המידע האמיתי, לדוגמא אפשר לבנות מודל ולאמן אותו על דאטה עם תמונות שמכילות אובייקטים שונים ואז המודל יוכל לדעת איזה אובייקט יש בתמונה חדשה שהוא לא "ראה" קודם.
למידה עמוקה היא תחום בתוך למידת מכונה שבו משתמשים במודלים יותר גדולים עם יותר שכבות ופרמטרים, בלמידה עמוקה בדרך כלל במקום לתת למודל את התכונות של ה data מביאים לו את ה raw data והמודל מנסה לבד להבין אילו תכונות בדאטה יותר חשובות ולפי זה לעשות את ה prediction.
אימון של מודל גדול עם הרבה שכבות ופרמטרים צורך הרבה משאבים כמו חומרה, חשמל וזמן, בנוסף כדי לאמן מודל גדול צריך הרבה לאסוף הרבה דאטה – כלומר צריך גם אנשים שיאספו ויסווגו את הדאטה.
בעיה נוספת ב deep learning זה איכות הדאטה: אם לדוגמא יש לנו דאטה של תמונות אז צריך שמישהו יעבור על הדאטה ויסווג איזה אובייקט יש בכל תמונה ויכול להיות שבחלק מהתמונות יש טעויות בסיווג או שהאובייקט בתמונה לא ברור והתמונה יותר תבלבל את המודל ותפריע לתהליך האימון שלו.

בפרויקט הזה ננסה לפתור את הבעיות האלו בעזרת חיתוך הדאטה:

- נממש מדדים שונים שיסווגו את הדאטה לפי רמת קושי

- נבדוק איזה חלק מהדאטה יותר משמעותי ללמידה: הדאטה הקל או הדאטה הקשה?
- נמצא את הדאטה הפחות איכותי (=רעש) ונוציא אותו מהדאטה

כך נוכל לאמן מודל בצורה יותר יעילה על data יותר איכותי ונחסוך במשאבים ואולי אפילו נשפר את האיכות של המודל.

2. תיאור הבעיה

בלמידת מכונה המודלים השתפרו בשנים האחרונות בעקבות בניית מודלים עם יותר שכבות ופרמטרים שאומנו על מערכי נתונים גדולים. בהרבה סוגים של רשתות נוירונים כמו זיהוי תמונה, שפה או דיבור השגיאה של המודל יורדת ככל שהמודל יותר גדול או כל שיש יותר דאטה בסדר גודל שלם, ולכן כדי להוריד את השגיאה מ 3% ל 2% לדוגמא, צריך לאסוף פי 10 יותר נתונים או לבנות מודל הרבה יותר גדול עם יותר שכבות ופרמטרים שהאימון שלו ידרוש הרבה יותר משאבים וזמן מה שמוביל לשאלה:

- האם צריך את כל הנתונים כדי להגיע לביצועים האלו?
- האם כל הדאטה נחוץ בשביל לאמן את המודל או שאפשר להגיע לאותם תוצאות רק עם חלק מהדאטה בלי לפגוע בביצועים?
- איך אפשר למצוא את הדאטה שיותר חשוב שהמודל יתאמן עליהם ולאמן את המודל דווקא עליהם וכך לחסוך כוח חישוב וזמן ואפילו לשפר את הביצועים של המודל בלי לאסוף כמויות אדירות של דאטה?
- אולי אפשר אפילו להגיע למודל עם accuracy יותר גבוה אם ניקח רק חלק מהדוגמאות

הפתרון ה"נאיבי": אפשר עבור כל דוגמת אימון לאמן 2 מודלים עם ובלי הדוגמא ואז לראות את ההבדל ב test loos וב accuracy.

בעיות בפתרון הנאיבי:

1. לא מעשי מבחינת זמן ריצה – עבור 1000 דוגמאות נצטרך להריץ 1000 מודלים
2. כל מודל מאותחל רנדומלית עם פרמטרים שונים ולכן כל מודל יסתכל שונה על כל דוגמת אימון ודוגמא שהייתה חשובה למודל אחד לא בטוח שתהיה חשובה למודל אחר

ולכן צריך למצוא אלגוריתם שייתן מידע על הדאטה בתנאים הבאים:

- המידע יינתן בשלב מוקדם של הלמידה
- היעילות של האימון (כולל ה preprocessing בשביל לקבל החלטה איזה דאטה לקצץ) מבחינת משאבים (זמן, כוח חישוב וכו') תשתפר בעקבות השימוש באלגוריתם
- ה accuracy של המודל שיקבל רק את הדאטה המשמעותי יהיה טוב לפחות כמו של מודל שיקבל את כל הדאטה

3. סקירת עבודות דומות בספרות והשוואה

- *An empirical study of example forgetting during deep neural network learning* [1]

המחקר מגדיר מדד לסיווג הדוגמאות אימון: אם מודל למד דוגמא מסוימת ונתן לה את המחלקה הנכונה ואז ב epoch הבא המודל נתן לה מחלקה לא נכונה אז המודל "שכח" אותה, במחקר מצאו שבעוד שיש דוגמאות שברגע שהמודל למד אותם הוא לא ישכח אותם יש דוגמאות שהמודל כל הזמן שוכח אותם, ולכן אפשר להגדיר מדד שככל שדוגמת אימון מסוימת תישכח יותר פעמים ככה היא יותר קשה, במחקר מצאו שדווקא התמונות שלא כל כך ברורות נשכחות יותר פעמים.

- *Deep learning on a data diet: Finding important examples early in training* [2]

המחקר מציע שיטה לחיתוך הדאטה EL2N – מדד שמימשנו בפרויקט והשווינו אות למדדים אחרים, הרעיון הוא לקחת 10 מודלים ולהריץ כל אחד מהם 10 epochs ואז לבדוק את הממוצע של L2 norm על וקטור השגיאה. במחקר הדגימו איך אפשר לקחת רק את הדוגמאות עם השגיאה הכי גדולה – 50% ו 75% בשביל cifar10 ו cifar100 בהתאמה בלי לפגוע ב accuracy.

- *Beyond neural scaling laws: beating power law scaling via data pruning* [3]

במחקר הזה מציעים 2 אלגוריתמים לסיווג הדאטה לפי דוגמאות קשות וקלות:

1. Student-teacher – מריצים מודל אחד 10 epochs ומודל שני epoch אחד ואז לפי ההפרש בין ה scores של שני המודלים אפשר להעריך אם הדוגמא קשה או קלה
2. K-means – מריצים אלגוריתם k-means על הדאטה לפני שמאמנים את המודל ובודקים עבור כל דוגמת אימון את המרחק שלה מהמרכז של המחלקה שאליה היא שייכת.

בנוסף במחקר הזה מצאו שאם יש הרבה דאטה אפשר לקצץ את הדוגמאות הקלות, אבל אם דווקא חסר דאטה עדיף לקצץ את הדוגמאות הקשות.

4. תיאור הפתרון

חלק 1: בחירת מודל ל cifar10/100

בפרויקט בחרנו לעבוד עם cifar10/100 עם מודל resnet18 והרצנו מספר ניסויים כדי לייצב 2 מודלים שיתנו תוצאות יחסית טובות על הדאטה ואז נוכל להריץ עליהם ניסויים בהמשך:

Cifar10 – 60000 תמונות RGB של 10 מחלקות, 6000 תמונות לכל מחלקה
Cifar100 – 60000 תמונות RGB של 100 מחלקות, 600 תמונות לכל מחלקה

כדי להגדיל את כמות הדאטה עשינו transforms:
- Random crop בגודל 4
- Random horizontal flip
- Random rotation עם מקסימום 15 מעלות

דוגמא לדאטה לפני ואחרי אפשר לראות כאן

Cifar10

עבור cifar10 השתמשנו במודל resnet18 + Adam(learning rate =0.001)

את ה learning rate הכפלנו ב 0.5 אחרי כל 2 epochs שה accuracy לא השתפר על ה test-set.
שינויים שעשינו במודל:
1. בקונבולוציה הראשונה שינינו את ה kernel size ל 3 x 3
2. ב fully connected בסוף המודל שינינו את גודל הפלט מ 1000 ל 10

Cifar100

עבור cifar100 השתמשנו במודל resnet18 + SGD(learning rate =0.001) + momentum(0.9)
את ה learning rate הכפלנו ב 0.3 ב 40, 45, 50 epochs

ה accuracy של resnet18 על cifar100 הוא ~65% לכן שינינו קצת את מבנה המודל כדי להגיע ל ~75% accuracy:
1. בקונבולוציה הראשונה שינינו את ה kernel size ל 3 x 3
2. ב fully connected בסוף המודל שינינו את גודל הפלט מ 1000 ל 100
3. ביטלנו את ה residual layer לגמרי מכיוון שהוא גם לא שיפר את המודל וגם הזמן ריצה שלו לקח פי 2
4. בשכבה השנייה והשלישית שינינו את ה stride ב conv2d מ 2 ל 1

חלק 2: סיווג הדוגמאות אימון לפי דרגת קושי

בחלק זה נממש מדדים שונים להערכת רמת הקושי של הדאטה (אפשר לראות כאן את הקורלציה בין המדדים).

Flip

בזמן האימון של המודל יש דוגמאות שלמודל ברור מה ה prediction שלהם מההתחלה לעומת דוגמאות שהמודל כל הזמן משנה את ה prediction שלהם ולכן אפשר להגדיר מדד שככל שה prediction משתנה יותר פעמים ככה הדוגמא יותר קשה ללימוד:

1. נאתחל מערך של counters בגודל הדאטה
2. נריץ מספר epochs ואחרי כל epoch נבדוק: אם ה prediction השתנה נעלה את ה counter של אותה דוגמה ב 1
3. הדוגמאות עם ה counter הכי גבוה הם הדוגמאות הקשות

Forget

המדד הוצע במאמר [1], בזמן האימון של המודל יש דוגמאות שהמודל למד מה המחלקה שלהם הוא לא "ישכח" אותם לעומת דוגמאות אחרות שאפילו אם המודל ילמד אותם הוא יכול "לשכוח" אותם ב epoch הבא ולכן נגדיר מדד כך שככל שדוגמא נשכחה יותר פעמים כך היא יותר קשה:

1. נאתחל מערך של counters בגודל הדאטה
2. נריץ מספר epochs ואחרי כל epoch נבדוק: עבור כל הדוגמאות שהמודל נתן להם prediction נכון ב epoch הקודם – אם ב epoch הנוכחי ה prediction השתנה נעלה את ה counter של אותה דוגמא ב 1
3. הדוגמאות עם ה counter הכי גבוה הם הדוגמאות הקשות

EL2N

המדד הוצע במאמר [2], סיווג קושי הלמידה של דוגמאות האימון לפי ממוצע L2 Norm על וקטור השגיאה של ה prediction של ה ensemble:

1. ניקח 10 מודלים שונים (מאותחלים רנדומלית עם פרמטרים שונים) ונאמן אותם 10 epochs
2. נעשה ממוצע על השגיאה של כל האלגוריתמים לפי - (EL2N) כלומר לכל מודל נעריך את השגיאה לפי L2 Norm ואז נעשה ממוצע על השגיאות של כל המודלים

חישוב EL2N:

$$E = size\ of\ ensemble$$

$$M = dataset\ size$$

$$C = number\ of\ classes$$

$$yOneHot[M][C] = true\ labels\ of\ dataset\ one\ hot\ encoded$$

$ensembleSoftmax[C][M][C] = softmax\ on\ prediction\ of\ 10\ models\ on\ the\ data$

$EL2N[M] = empty\ vector$

$for\ m \in 0 \dots M-1:$

$\quad for\ e \in 0 \dots E-1:$

$$L2Scores = \sqrt[2]{\sum_{c=0}^{C-1}(yOneHot[m][c] - ensembleSoftmax[e][m][c])^2}$$

$$EL2N[m] = \frac{\sum L2Scores}{E}$$

3. לכל דוגמת אימון נעריך כמה היא קשה ללמידה לפי הציון של EL2N: אם הוא גדול אז זה אומר שההפרש בין ה prediction של ה ensebmle ל true labels גדול מידי גם אחרי 10 epoches ולכן נסווג את הדוגמא כקשה לאימון

STD

בדיקת סטיית התקן של ה prediction של ensemble של אלגוריתמים שונים לכל דוגמת אימון:

1. נאמן 10 אלגוריתמים שונים על הדאטה כמו ב EL2N
2. לכל דוגמת אימון נבדוק את השונות לכל class בין המודלים השונים - אם יש שונות גבוהה אז כנראה שמדובר בדוגמא קשה ולכן כל מודל נותן ציון אחר ואם יש שונות נמוכה אז כל המודלים חשבו אותו דבר ואז כניראה שמדובר בדוגמא קלה

חישוב STD:

$E = size\ of\ ensemble$

$M = dataset\ size$

$C = number\ of\ classes$

$yOneHot[M][C] = true\ labels\ of\ dataset\ one\ hot\ encoded$

$ensembleSoftmax[C][M][C] = softmax\ on\ prediction\ of\ 10\ models\ on\ the\ data$

$ensembleStd[M] = empty\ vector$

$for\ m \in 0 \dots M-1:$

$$ensembleStd[m] = \sum std(ensembleSoftmax[0 \dots E-1, m, 0 \dots C-1], dim = 0)$$

True prediction sum

1. נאמן 10 אלגוריתמים שונים על הדאטה
2. נבדוק לכל מודל כמה מודלים מתוך ה ensemble צדקו עליו - ככל שיותר מודלים צדקו על הדוגמא ככה היא יותר קלה ללמידה וההפך

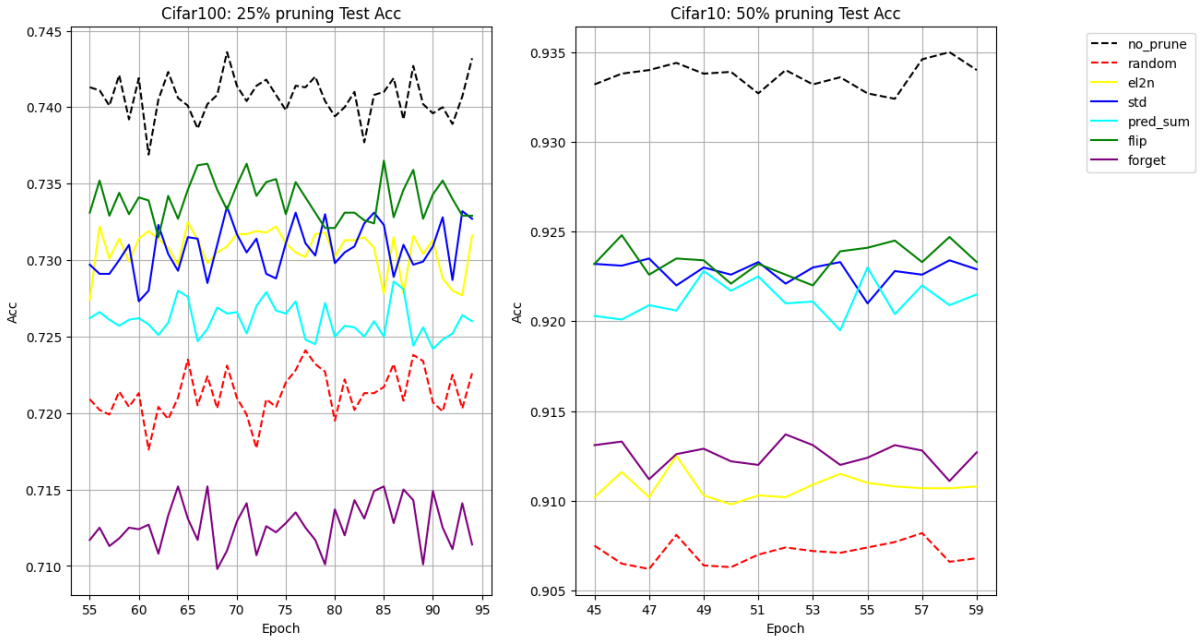
חלק 3: סדרת ניסויים על המדדים

בחלק זה ננסה להשוות בין המדדים השונים על ידי סידרת ניסויים, בכל הניסויים עשיתי השוואה למודל שאומן על כל הדאטה (הקו המקווקו השחור) ולמודל שבו חתכנו מהדאטה רנדומלית (הקו המקווקו האדום).

ניסוי 1: הורדת הדוגמאות הקלות מה dataset

בניסוי הזה הורדנו 25% מהדוגמאות הקלות מ cifar100 ו 50% מ cifar10 ובדקנו כמה זה משפיע על המודל, ציר ה x תואם למספר ה epoch – החל מה epoch שבו המודל התייצב וציר ה y תואם ל accuracy.

אפשר לראות שאפשר להגיע כמעט לאותו תוצאה אפילו עם חיתוך של 25% מהדוגמאות הקלות ביותר ב cifar100 או 50% ב cifar 10, ודווקא אם חותכים בצורה רנדומלית מגיעים לתוצאות פחות טובות.



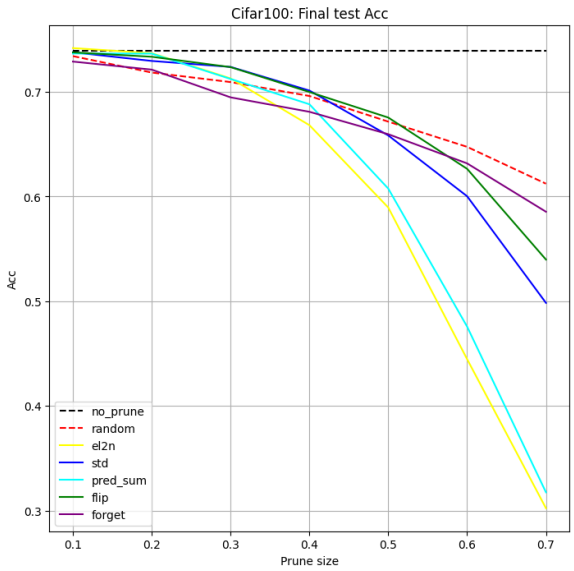
ניסוי 2: חיתוך של בין 0% ל 70% מהדוגמאות הקלות

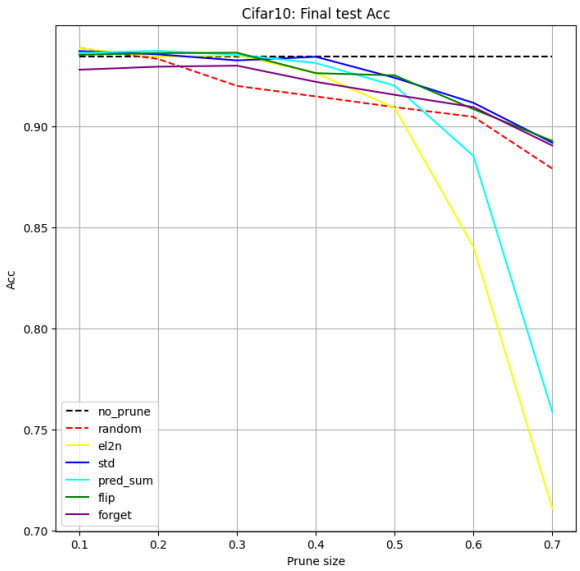
בניסוי הזה בדקנו איך חיתוך הדאטה לפי המדדים השונים משפיע על המודל ככל שמורידים יותר דוגמאות קלות.

ציר ה x תואם לגודל החיתוך של הדוגמאות הקלות לדוגמא אם x = 0.1 אז המודל אומן על 90% מהדוגמאות הקשות אחרי שקיצצנו את ה 10% הכי קלות וציר ה y תואם לתוצאה הסופית של המודל על ה test-set.

מסקנות מהניסוי:

1. אפשר לראות שאפילו שהשתמשנו ב cifar100 שבו יש 60000 דוגמאות ו 100 מחלקות כלומר 600 תמונות למחלקה ואין בו מספיק דאטה - בכל זאת אפשר לחתוך 20% מהדאטה בלי לפגוע ב accuracy ואפילו אם נוריד אחוז יותר גבוה מהקלים (עד 35%) עדיין נקבל תוצאות יותר טובות מאשר חיתוך רנדומלי, מה שמוכיח שאפשר כבר בשלב מוקדם של הלמידה לדעת אילו דוגמאות יותר חשובות למודל ואיזה פחות.
2. בנוסף אפשר לראות שמתי שחתכנו יותר מידי מהדוגמאות הקלות המודל למד יותר גרוע מאשר חיתוך רנדומלי, במאמר [2] הסבירו שכנראה מתי שהמודל רואה רק את הדוגמאות הקשות אז הוא מתאים את עצמו דווקא לקבוצה קטנה של דוגמאות קשות ולא למגוון של דוגמאות.





האינטגרל על השטח מתחת לגרף מנורמל לפי no pruning:

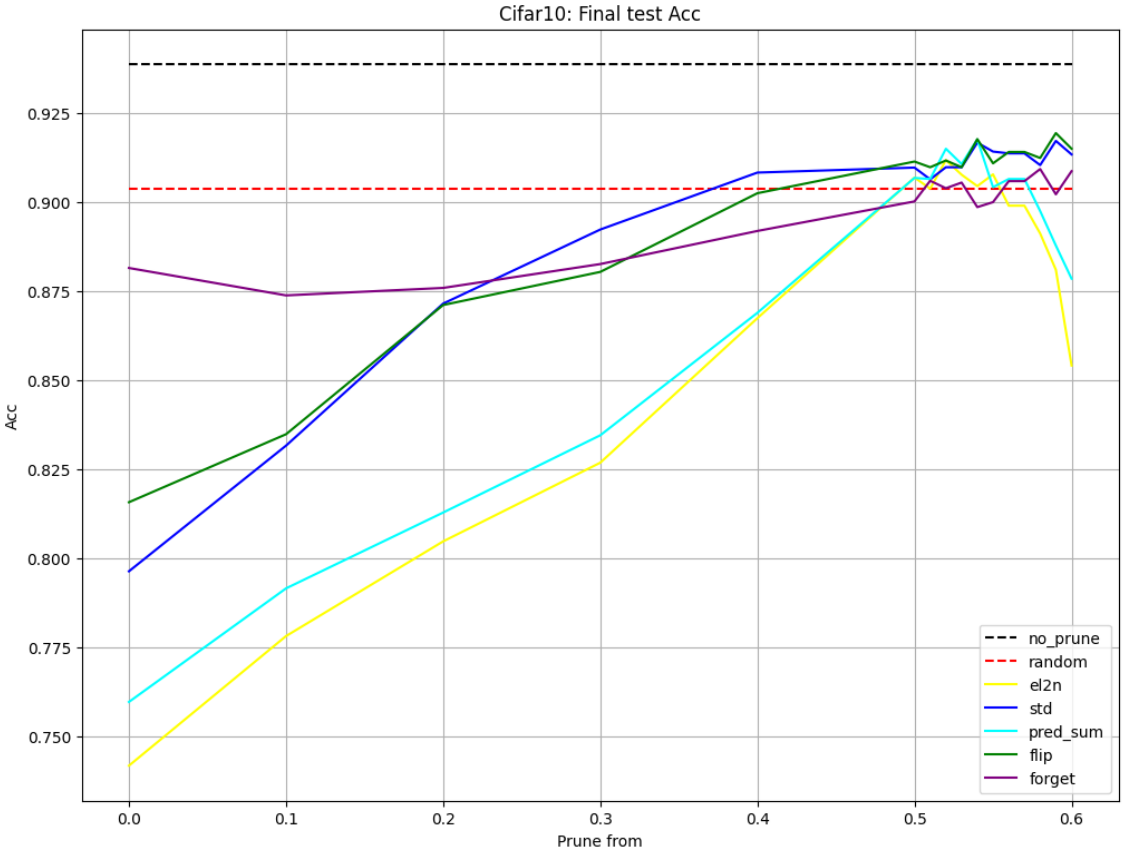
	random	EL2N	STD	Pred sum	flip	forget
Cifar10	0.98	0.96	0.991	0.976	0.989	0.984
Cifar100	0.928	0.83	0.911	0.849	0.925	0.913

ניסוי 3: אימון של sliding window בגודל 40% על הדאטה

בגרף הבא:

תוצאה סופית על ה test-set אחרי אימון של sliding window בגודל 40% מהדאטה, ציר ה x תואם לנקודה ממנה התחלנו לקחת את ה 40%: מה 40% הכי קלים בצד שמאל עד ל 40% הכי קשים בצד ימין לפי המדדים השונים וציר ה y תואם לתוצאה הסופית של המודל על ה test-set.

לדוגמא אם בציר ה x הערך הוא 0.3 אז לקחנו את הדאטה בטווח של 30%-70% מהדוגמאות לפי רמת הקושי של כל מדד.



מסקנות מהניסוי:

1. בניסוי הזה אפשר לראות שככול שניקח דוגמאות יותר קשות - ככה המודל יגיע לתוצאה יותר טובה ושבלי דוגמאות קשות המודל לא ילמד יותר טוב מאשר בחיתוך רנדומלי כי בחיתוך רנדומלי יש גם דוגמאות קשות
2. אם לוקחים דווקא את הדוגמאות הקשות (אבל לא הכי קשות) אז המודל לומד הרבה יותר טוב מאשר random pruning

בנוסף אפשר לראות במדדים EL2N ו pred sum שמתי שלוקחים דווקא את הדוגמאות הכי קשות אז המודל לומד קצת פחות טוב, במאמר [2] הסבירו שהדוגמאות הכי קשות יכולות להיות רעש ומידע לא טוב ולכן דווקא הדוגמאות הכי קשות לא טובות למודל, כלומר יש לנו פה דרך לזהות את הרעשים בדאטה.

5. מה עשינו עד כה?

- ✓ מציאת מודל ל cifar10/100
- ✓ הרצת ensemble של מודלים וחילוץ המדדים EL2N, pred sum ו STD מתוך ה ensemble
- ✓ אימון מודל על cifar10/100 והוצאת המדדים flip, forget
- ✓ ניסוי 1: חיתוך 25% מהקלים ב cifar100 ו 50% מהקלים ב cifar10 והשוואה בין הביצועים של המודלים לפי השיטות השונות
- ✓ ניסוי 2: חיתוך בטווח של 0%-70% מהדאטה והשוואה בין הביצועים של המודלים השונים
- ✓ ניסוי 3: אימון של 40% מהדאטה מהקלים לקשים ובדיקה איך כל חלק מהדאטה משפיע על תהליך הלמידה של המודל לפי המדדים השונים
- ☒ מימוש student-teacher מודל לסיווג הדאטה
- ☒ מימוש k-means מודל לסיווג הדאטה
- ☒ השוואה בין המדדים student-teacher ו k-means לשאר המדדים

6. נספחים

א. טרנספורמציות על הדאטה.

בשורה הראשונה התמונה המקורית ובשורה השנייה התמונה אחרי טרנספורמציות

ב. קורלציה בין המדדים השונים.

Cifar100	**EL2N**	**STD**	**True pred sum**	**Flip**	**Forget**
EL2N	1	0.763640	0.931154	0.631182	0.015292
STD	0.763640	1	0.653038	0.667967	0.121611
True pred sum	0.931154	0.653038	1	0.544786	0.114446
Flip	0.631182	0.667967	0.544786	1	0.394457
Forget	0.015292	0.121611	0.114446	0.394457	1

Cifar10	**EL2N**	**STD**	**True pred sum**	**Flip**	**Forget**
EL2N	1	0.780107	0.944299	0.658627	0.399751
STD	0.780107	1	0.660269	0.586532	0.377763
True pred sum	0.944299	0.660269	1	0.627331	0.350162
Flip	0.658627	0.586532	0.627331	1	0.770378
Forget	0.399751	0.377763	0.350162	0.770378	1

מהקורלציה אפשר לראות:

- יש קורלציה גבוהה בין המדדים חוץ מה forget ולכן דוגמת אימון שמדד אחד יסווג אותה כקשה תהיה בסבירות גבוהה קשה גם לפי המדדים האחרים, כלומר אפשר להבין שקושי של מודל ללמוד דוגמא מסוימת זה תכונה של הדאטה ולא ספציפי למודל או מדד מסוים.
הסיבה שלמדד forget יש קורלציה נמוכה היא בגלל שברוב דוגמאות האימון אחרי שמודל למד את הדוגמא הוא לא שוכח אותה ולכן גם אם הדוגמא לא הכי קלה ה Forget שלה יהיה 0

- ל EL2N ול pred sum יש קורלציה יחסית גבוהה ובעצם אפשר להגדיר את EL2N כמדד רך של pred sum בגלל ש EL2N במקום לבדוק כמה מודלים צדקו על דוגמא מסוימת (מספר שלם) הוא בודק איזה אחוז כל מודל נותן לכל דוגמא עבור כל המחלקות
- למרות שיש קורלציה גבוהה בין רוב המדדים אפשר לראות שיש ביניהם הבדלים ויכול להיות שמדד מסוים ייתן תוצאות יותר טובות מאחרים בסיווג דוגמאות או במציאת רעשים בדאטה כמו שרואים בחלק של הניסויים

ג. תכנון הפרויקט – ברזולוציה של שבועיים.

תאריך	
22.1.2023	הגשת דוח אלפא
16.3.2023	הגשת לוז עבודה לפרויקט
23.3.2023	לימוד המאמר "Beyond neural scaling laws: beating power law scaling via data pruning" [3]
6.4.2023	מימוש האלגוריתם student-teacher וסיווג בעזרת k-means
20.4.2023	וניסויים על המדדים החדשים תוך השוואה למדדים קודמים
4.5.2023	הגשת דוח beta
18.5.2023	מסקנות מהניסויים וניסיון על דאטה "אמיתי"
1.6.2023	מסקנות סופיות מהפרויקט ותחילת כתיבת דוח סופי ופרזנטציה
15.6.2023	המשך כתיבת דוח סופי ופרזנטציה
24.6.2023	הגשת דוח סופי
17.7.2023	פרזנטציה
31.7.2023	הצגה בכנס פרויקטים

ד. רשימת ספרות / ביבליוגרפיה

[1] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. In ICLR, 2019. link

[2] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. Adv. Neural Inf. Process. Syst. 34, December 2021. link

[3] Sorscher, Ben, et al. "Beyond neural scaling laws: beating power law scaling via data pruning." arXiv preprint arXiv:2206.14486 (2022). link