סקירה זו היא חלק מפינה קבועה בה אני סוקר מאמרים חשובים בתחום ה-ML/DL, וכותב גרסה פשוטה וברורה יותר שלהם בעברית. במידה ותרצו לקרוא את המאמרים הנוספים שסיכמתי, אתם מוזמנים לבדוק את העמוד שמרכז אותם תחת השם deepnightlearners.

לילה טוב חברים, היום אנחנו שוב בפינתנו deepnightlearners עם סקירה של מאמר בתחום הלמידה העמוקה. היום בחרתי לסקירה את המאמר שנקרא:

Meta-Learning Requires Meta-Augmentation

פינת הסוקר:

המלצת קריאה ממייק: מומלץ לאוהבי מטה-למידה אך לא חובה

בהירות כתיבה: גבוהה

רמת היכרות עם כלים מתמטיים וטכניקות של ML/DL הנדרשים להבנת מאמר: נדרשת הבנה טובה של מושגי יסוד של תמום מטה-למידה (meta-learning).

יישומים פרקטיים אפשריים: שיפור ביצועים במשימות של מטה-למידה באמצעות אוגמנטציה של לייבלים.

פרטי מאמר:

לינק למאמר: זמין להורדה.

לינק לקוד: זמין כאן

פורסם בתאריך: 04.11.21, בארקיב.

הוצג בכנס: NeurIPS2020

תחום מאמר:

- שיטות אוגמנטציה למטה-למידה (meta-learning)
- שיטות התמודדת עם אוברפיטינג (overfitting) במטה-למידה

כלים ומושגים מתמטיים במאמר:

- אפיזודה של משימת מטה-למידה
- למידה N-way, K-shot
- זיכרון (memorization) במשימות מטה-למידה
- אנטרופיה מותנית (conditional entropy - CE)
- אוגמנטציה שומרת CE

תמצית מאמר:

המאמר הנסקר מציע שיטה חדשה לאוגמנציה שבאה להתמודד עם בעיית אוברפיטינג (overfitting), המתרחשת במשימות מטה-למידה. המאמר מציע לבצע אוגמנטציה פסאדו-אקראית ללייבלים (ולא לדאטה!!) של המשימות של base learner (מודל חיצוני) ואותה אוגמנטציה גם ללייבלים של המשימות של מודל פנימי. בדרך זו מודל פנימי יהיה "חייב" לשחזר את האוגמנטציה שהשתמשו בה במודל חיצוני וכבר לא יכול "להתעלם" מהעדכונים שלו שלטענת המאמר מסייע להתגבר על אוברפיטינג במשימות מטה-למידה.

תקציר מאמר:

נתחיל מלהיזכר מה זה בעיית מטה-למידה:

מה זה בעיית מטה-למידה:

כמו שכולכם יודעים בכל בעיית למידה supervised נתון לנו סט אימון (X, Y), המכיל זוגות של דוגמאות והלייבלים שלהם (תיוגים). המטרה של אימון supervised היא למדל את הפונקציה הממפה X ל-Y.

לעומת למידה supervised בבעיית מטה-למידה יש לנו מספר משימות T_i, כאשר כל משימה מורכבת מסט תומך (support set), המכיל כמה זוגות של דוגמאות והלייבלים (x_s, y_s) וסט שאילתה (query set) (x_q,y_q), שביחד בונים **אפיזודה**. נציין שבדרך כלל גם סט תומך וגם סט שאילתה מכילים מספר מאוד קטן של דוגמאות. בנוסף נתונים לנו **סט אימון מטה** (meta train set), המקביל לסט אימון בבעיית ML רגילה ו**מטה-טסט סט** (כמו טסט סט ב-ML רגיל), המכילים כמה אפיזודות כל אחד. המטרה של מטה-למידה היא לאמן מודל (הנקרא base learner או מודל חיצוני) על הדאטה שבסט התומך (x_s, y_s) כאשר הפלט שלו הינו המודל לחיזוי y_q מ-x_q מתוך סט השאילתה. כלומר המטרה של מטה-למידה היא להקנות למודל החיצוני יכולת "ללמד" את המודל הפנימי (learner).

במודלי מטה-למידה יש שני שלבי אימון: **השלב הפנימי** שבו מודל חיצוני מעדכן את מודל פנימי במטרה לשפר את יכולת החיזוי שלו עבור דוגמאות מסט שאילתה x_q ובמסגרת **השלב החיצוני** מעדכנים מודל חיצוני עצמו במטרה לשפר את יכולתו "ללמד" מודל פנימי. יש כמה סוגים של שיטות מטה-למידה ואחת מהנפוצות מהם היא MAML. ב-MAML המודל החיצוני הוא רשת נוירונים שמאמנים אותה בשביל לעדכן את המשקלים של המודל הפנימי שהוא גם כן רשת נוירונים.

אוגמנטציה: כידוע המטרה העיקרית של אוגמנטציה של דאטה במשימות ML היא מניעת אוברפיטינג ע"י יצירה של דוגמאות נוספות לאימון של מודל. לאור זה נתאר עתה את סוגי האוברפיטינג המתרחשים במשימות מטה-למידה.

סוגי אוברפיטינג במודלי מטה-למידה:

יש שני סוגים עיקריים של אוברפיטינג שעלולים להתרחש במהלך אימון של מודלי מטה-למידה:

- **זיכרון (memorization)** - מודל פנימי מתעלם מהעדכונים שמודל חיצוני מעביר לו ומשתמש בפועל רק בדוגמאות שלה מסט השאילתה (לא קיימת בבעיות ML רגילות). למעשה במקרה הזה לסט תומך אין שום השפעה על חיזוי של מודל פנימי עבור דוגמאות מסט שאילתה.

- **אוברפיטינג של learner** - מודל חיצוני עושה אוברפיטינג על סט אימון מטה ואינו מצליח להכליל למטה-טסט סט (זה הסוג הרגיל של אוברפיטינג הקורה במשימות ML סטנדרטיות).

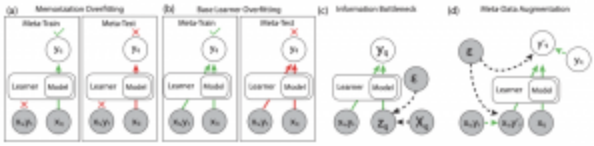

Figure 1: Meta-learning problems provide support inputs (x_s, y_s) to a *base learner*, which applies an update to a *model*. Once applied, the model is given query input x_q, and must learn to predict query target y_q. **(a)** Memorization overfitting occurs when the base learner and (x_s, y_s) does not impact the model's prediction of y_q. **(b)** Learner overfitting occurs when the model and base learner leverage both (x_s, y_s) and x_q to predict y_q, but fails to generalize to the meta-test set. **(c)** Yin et al. [37] propose an information bottleneck constraint on the model capacity to reduce memorization overfitting. **(d)** To tackle both forms of overfitting, we view meta-data augmentation as widening the task distribution, by encoding additional random bits ϵ in (x_s, y_s) that must be decoded by the base learner and model in order to predict a transformed y_q'.

בשביל להבין באלו סוגים של משימות מתרחשת אוברפיטינג מסוג זיכרון אנו צריכים להגדיר את המושג החשוב הבא:

הגדרה: סט משימות נקרא mutually exclusive(Mex) כאשר מודל אחד לא יכול לפתור את כל המשימות ביחד.

למשל אם במשימה אחת מסט המשימות יש תמונות של סוסים מתויגות עם לייבל 0 ותמונות של כלבים המתויגות עם לייבל 1 ובמשימה השנייה הסוס מקבל לייבל 1 והכלב מקבל לייבל 0, לא קיים מודל שיכול ללמוד אותה את שתי משימות אלו יחד. יש מחקרים שטוענים שסטים משימות Mex הם יותר קלים בתחום מטה-למידה כי מודל פנימי "חייב" לנצל מידע מסט תומך (x_s, y_s) כדי לבצע את המשימה שלה. כנראה הסיבה לכך היא שהמודל יתקשה גם "לזכרן" את המשימה מהסט התומך, ובאותו זמן ללמוד משימה "מנוגדת" למשימה זו מהסט התומך. ההנחה היא שהמודל "ייאלץ" ללמוד "פיצ'רים מועילים" מהדוגמאות מהסט התומך שינוצלו לאחר מכן ע"י המודל במהלך אימון על סט השאילתה.

לעומת זאת אם סט המשימות אינו מקיים את תכונת MeX, אוברפיטינג מסוג זיכרון עלול להתרחש (לטענת המאמר) כי מודל אחד כן יכול ללמוד לחזות y_q רק על בסיס x_q בלי להסתמך על מידע מ-(x_s, y_s). כאשר זה קורה הביצועים של מודל מטה-למידה טובים על סט אימון מטה וסופגים ירידה משמעותית על מטה טסט סט (מטה-הכללה גרוע). הסיבה לכך היא שהמודל החיצוני פשוט "מזכרן" את הסט התומך במקום "לנצלו בשביל ללמוד איך ללמד את המודל הפנימי".

צריך לציין שרוב המשימות מטה-למידה מסוג סיווג N-way, K-shot (מספר הדוגמאות בכל סט תומך של משימה הינו K ויש בכל משימה N לייבלים שנדגמים באקראי), הסטים של המשימות הינם MeX כי אנחנו דוגמים אפיזודות באופן רנדומלי כך שכל קטגוריה מקבלת לייבל שונה בכל משימה. כלומר

במשימה מסוימת החתול יכול לקבל לייבל 0 כאשר במשימה אחרת הוא יקבל לייבל 1. כאשר המשימות הן מסוג רגרסיה העניינים מסתבכים וסטים של משימות מתקשות לקיים את MeX. כדי להתגבר על בעיות הזיכרון במקרים האלו ניתן להגביל את הזרימה של המידע בין x_q ל- y_q (דרך המידע ההדדי שלהם) אבל צריך לעשות את זה בעדינות בשביל לא להגיע למצב של underfitting.

הסוג השני של אוברפיטינג (learner overfitting) קורה כאשר המודל החיצוני מצליח את הדאטה שלו (x_s,y_s) בשביל לעזור למשימות של המודל הפנימי בסט אימון מטה אבל אינו מצליח להכליל את זה לאפיזודות של מטה-טסט סט.

אוקיי ,אז איך מתמודדים עם אוברפיטינג מהסוג הראשון בלי להגביל את זרימת המידע בין x_q ל- y? בדומה ללמידה הרגילה התשובה היא - אוגמנטציה של דאטה. אבל לא האוגמנטציה הרגילה של הדוגמאות אלא אוגמנטציה של הלייבלים. במאמר קוראים לזה מטה-אוגמנטציה.

מטה-אוגמנטציה:

בשביל להבין את הרעיון של מטה- אוגמנטציה בואו קודם נבין איזה סוגי אוגמנטציה אפשר לעשות לדאטה. קודם כל פעולת אוגמנטציה ניתן להגדיר בתור מיפוי (X',Y')->(X,Y):F. אוגמנטציה נקראת שומרת אנטרופיה מותנית (CE preserving) כאשר האנטרופיה של לייבל שעבר אוגמנטציה בהינתן הדוגמא שעברה אוגמנטציה, שווה לאנטרופיה המותנית של הלייבל המקורי בהינתן הדוגמא המקורית: H(Y'|X') = (H(Y|X. למשל אוגמנטציה מסוג סיבוב של תמונה תוך שמירה על אותו הלייבל הינה שומרת אנטרופיה מותנית. כמו כן אוגמנטציה נקראת מגדילה אנטרופיה מותנית (CE-increasing) כאשר האנטוריה המותנית עולה לאחר אוגמנטציה. למשל אם נעשה אוגמנטציה רק ללייבל של תמונה נתונה (נוסיף אליו איזה מספר נגיד) אז האנטרופיה המותנית תעלה כי לאותה תמונה יהיו שני לייבלים שונים.

אז המאמר אומר דבר כזה: אנו צריכים אוגמנטציה שתקשר את הזוגות (x_s, y_s) לזוגות (x_q, y_q) כך שמודל פנימי לא יוכל להביא את הלוס על סט השאילתה למינימום ע"י שימוש ב-x_q בלבד אלא "נכריח" אותו "לשתף פעולה" עם x_s. הדרך לעשות זאת היא לעשות אוגמנטציה שהיא CE-increasing למשימות. כלומר לכל משימה הלייבלים y_s ו- y_q "יעוותו" באותה צורה (יעברו "הצפנה" עם אותו מפתח שנבחר רנדומלית או אותה דגימה של רעש). במקרה הזה רשת פנימית יכולה לחזות את y_q המעוות מ-x_q רק אם היא הצליחה לפענח את מפתח ההצפנה (רעש פסאודו רנדומלי) שהוא יכול ללמוד רק מ-(x_s, y_s) המוצפן.

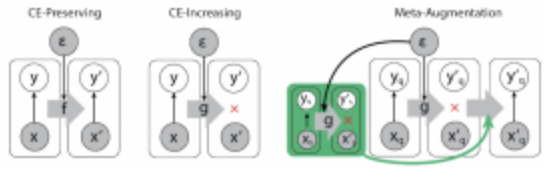

Figure 2: **Meta-augmentation:** We introduce the notion of *CE-preserving* and *CE-increasing* augmentations to explain why meta augmentation differs from standard data augmentation. Given random variables X, Y and an external source of random bits ϵ, we augment with a mapping $f(\epsilon, X, Y) = (X', Y')$. **Left:** An augmentation is *CE-preserving* if it preserves conditional entropy between x, y. **Center:** A *CE-increasing* augmentation increases $\mathcal{H}(Y'|X')$. **Right:** Invertible CE-increasing augmentations can be used to combat memorization overfitting: the model must rely on the base learner to implicitly recover ϵ from x'_s, y'_s in order to restore predictiveness between the input and label.

אינטואיציה לשיטה המוצעת:

אם ניקח משימה מסוימת (אפיזודה) וניצור סט מספיק גדול של משימות מאוגמנטות עם אותו מקור של רעש Δ, אז האנטרופיה המותנית של המשימה המוצפנת של המודל הפנימי תעלה ב-$H(\Delta)$. לכן בשביל לבצע את המשימה המודל הפנימי חייב להקטין את האנטרפיה הזאת באותה באמצעות "הלמידה" מהמודל החיצוני.

הישגי מאמר:

המאמר מראה שיפור בביצועים במשימות k-shot, N-way על מספר דאטהסטים המקובלים בתחום מטה-למידה. המחברים הצליחו להקטין את ההשפעה השלילית של אוברפיטינג מסוג זיכרון בתרחישים שבהם סט המשימות אינו MeX. המאמר משתמש ב-MAML בשביל לאמן את המטה-מודל שלהם. צריך לציין שעבור בעיות סיווג k-shot, N-way המחברים יצרו אפיזודות כך שסט המשימות שלהם הוא Non-MeX (למרות ש-k-shot, N-way קלאסי הוא כן MeX).

Figure 3: **Non-mutually-exclusive, intrashuffle, and intershuffle.** In this example, the dataset has 4 classes, and the model is a 2-way classifier. In non-mutually-exclusive, the model always sees one of 2 tasks. In intrashuffle, the model sees permutations of the classes in the non-mutually-exclusive tasks, which changes class order. In intershuffle, the model sees $4 \times 3 = 12$ tasks.

Figure 4: Mini-ImageNet results with MAML. **Left:** In a non-mutually-exclusive setting, this model exhibits memorization overfitting. Train-time performance is high, even before the base learner updates the model based on $\langle x_v, y_v \rangle$, indicating the model pays little attention to $\langle x_v, y_v \rangle$. The model fails to generalize to the held-out validation set. **Center:** This model exhibits learner overfitting. The gap between train pre-update and train post-update indicates the model does pay attention to $\langle x_v, y_v \rangle$, but the entire system overfits and does poorly on the validation set. The only difference between the left and center plots is the random seed. **Right:** With intershuffle augmentation, the gap between train pre-update and train post-update indicates the model pays attention to $\langle x_v, y_v \rangle$, and higher train time performance lines up with better validation set performance, indicating less overfitting.

דאטהסטים: Omniglot, Mini ImageNet, D'Claw, Pascal3D, Pose Regression.

נ.ב. אהבתי את החשיבה של מחברי המאמר. המאמר קריא, הרעיון מאוד אינטואיטיבי ומוסבר בצורה יפה.

#deepnightlearners

הפוסט נכתב על ידי מיכאל (מייק) ארליכסון, PhD, Michael Erlihson.

מיכאל עובד בחברת הסייבר Salt Security בתור Principal Data Scientist. מיכאל חוקר ופועל בתחום הלמידה העמוקה, ולצד זאת מרצה ומנגיש את החומרים המדעיים לקהל הרחב.