

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 שיצא לפני כ 3 חודשים.

תחום מאמר: שיטות אוגמנטציה למטה-למידה, שיטות התמודדות עם אוורייטינג במטה-למידה

תמצית מאמר: המאמר מציע שיטה חדשה לאוגמנטציה שבאה להתמודד עם בעיות אוברפיטינג המתרחשות במשימות מטה-למידה. המאמר מציע לבצע אותה (!!) אוגמנטציה ללייבלים (לא לדאטה!!) על המשימות של base learner (מודל חיצוני) ולייבלים של המודל הפנימי כך שהמודל הפנימי יהיה "חייב" לשחזר את האוגמנטציה שהשתמש בה המודל חיצוני וכבר לא יכול להתעלם מהעדכונים שלו שלטענת המאמר מסייע להתגבר על אוברפיטינג במודלי מטה-למידה

תקציר מאמר: נתחיל מלהגדיר מה זה בעיית מטה למידה:

מה זה מטה-למידה: כמו שכולכם יודעים בכל בעיית למידה supervised יש לנו טריין סט (X, Y) המכיל דוגמאות ולייבלים שנדגמו ממשימה מסוימת T כאשר המטרה ללמוד פונקציה מ X ל Y . במטה-למידה יש לנו מספר משימות T_i כאשר כל משימה מורכבת מסט תומך (support set) המכיל מכמה דוגמאות (x_s, y_s) זה סט שאילתה (query set) שביחד נקראים אפיזודה. מטה-טריין סט (מחליף טריין סט בבעיית ML רגילה) ומטה-טסט סט (כמו טסט סט ב- ML רגיל) מכיל כמה אפיזודות כל אחד. המטרה כאן היא לאמן רשת (הנקראת base learner או מודל חיצוני) על הדאטה שבסט התומך (x_s, y_s) כשהפלט שלו הינו המודל לחיזוי y_q מ- x_q מסט השאילתה. כלומר המטרה כאן היא להקנות למודל החיצוני יכולת ללמד את המודל הפנימי (learner).

במודלי מטה-למידה יש שני שלבי אימון: השלב הפנימי שבה הרשת החיצונית מעדכנת את המודל הפנימי בשביל לשפר את החיזוי של על x_q והשלב החיצוני במסגרתו מעדכנים את המודל החיצוני עצמה בשביל שהיא תדע "ללמד יותר טוב" את המודל הפנימי. יש כמה סוגים של מודלים ושיטות למטה-למידה ואחד מהנפוצות הם זה MAML שבו מאמנים את המודל החיצוני זה רשת שמאמנים אותה בשביל לעדכן את המשקלים של הרשת הפנימית (סקרתי את ההכללה של שיטה זה הנקראת iMAML באחד הפוסטים הקודמים שלי)

סוגי אוברפיטינג במודלי מטה-למידה: יש שני סוגים של אוברפיטינג שעלולים להתרחש במודלי מטה-למידה:

- א. זיכרון - (memorization) המודל הפנימי מתעלם מהעדכונים שהמודל החיצוני מעביר לו ומשתמש בפועל רק בדוגמאות שלה (לא קיימת בבעיות ML רגילות)
- ב. אוברפיטינג של learner המודל החיצוני עושה אוברפיטינג על מטה-טריין סט ואיננו מצליח להכליל למטה-טסט סט (סוג זה הינו הסוג הרגיל של אוברפיטינג הקורה בלמידה הרגילה)

בשביל להבין באילו סוגים של משימות מתרחשת אוברפיטינג מסוג זיכרון אנו צריכים את המושג החשוב הבא: סט משימות נקרא mutually exclusive (Mex) כאשר מודל אחד לא יכול לפתור את כולם ביחד (!!). למשל אם במשימה הראשונה תמונות סוס מתויגות עם לייבל 0 ותמונות של כלב מתויגות עם לייבל 1 ובמשימה השנייה הסוס מקבל 1 והכלב מקבל 0, אין מודל אחד שיכול ללמוד אותה. סטים כאלו הם יותר קלים למטה-למידה כי המודל הפנימי במקרה הזה חייב לנצל (x_s, y_s) כדי לבצע את המשימה שלה. כאשר סט המשימות אינו MeX אוברפיטינג מסוג זיכרון יכול להתרחש (לטענת המאמר) כי מודל אחד יכול ללמוד לחזות את y_q רק על בסיס x_q בלי להסתמך על (x_s, y_s) . כאשר זה קורה הביצועים של מודל מטה-למידה טובים על מטה-טריין סט וסופגים ירידה משמעותית המטה טריין סט (מטה-הכללה גרוע). הסיבה היא שהמודל החיצוני פשוט "מזכרן" את הטסט התומך במקום "ללמוד איך ללמד את המודל הפנימי".

צריך לציין שרוב המשימות מטה-למידה סיווג N-way, K-shot (מספר הדוגמאות בכל סט תומך של משימה היו K ויש בכל משימה N לייבלים שנדגמים באקראי) הסטים של המשימות הינם MeX כי אנחנו דוגמים את המשימות באופן רנדומלי כך שכל קלאס מקבל 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 והסטים של המשימות מתקשות לקיים את דרישות ה-MeX. כדי להתגבר על בעיות הזיכרון במקרים האלו ניתן להגביל את הזרימה של המידע בין x_q ל- y_q (דרך המידע ההדדי) אבל צריך לעשות את זה בעדינות בשביל לא להגיע ל-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 במטה-טריין סט 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_q . בדומה ללמידה הרגילה התשובה היא - אוגמנטציה של הדאטה. אבל לא האוגמנטציה רגילה 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 אוגמנטציה ניתן להגדיר כמיפוי $(X', Y) \rightarrow (X, Y)$: F. אוגמנטציה נקראת שומרת אנטרופיה מותנית (CE preserving) כאשר האנטרופיה של הלייבל שעבר אוגמנטציה בהינתן הדוגמא שעברה אוגמנטציה, שווה לאנטרופיה המותנית של הלייבל המקורי בהינתן הדוגמא המקורית: $H(Y|X) = H(Y'|X')$. למשל אוגמנטציה סיבוב של התמונה תוך שמירה על אותו הלייבל הינה שומרת אנטרופיה מותנית. כמו כן האוגמנטציה נקראת מגדילה אנטרופיה מותנית (CE-increasing) כאשר האנטרופיה 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) ו- (x_q, y_q) כך שהמודל הפנימי לא יכול להביא את טריין-לוס 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 הרשת הפנימית יכולה לחזות את y_q המעוותת מ- x_q רק דרך ההבנה את מפתח ההצפנה (רעש פסאודו רנדומלי) שהוא יכול ללמוד רק מ (x_s, y_s) המוצפן.

אינטואיציה לשיטה המוצעת: אז 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 עולה ב- $H(\Delta)$. אז בשביל לבצע את המשימה היא חייבת להקטין את האנטרופיה הזאת באותה כמות ע"י הפקתה המודל החיצוני

הישגי מאמר: הם מוכיחים על דאטה סטים שונים המקובלים בתחום של המטה-למידה שהם מצליחים להקטין את האפקט השלילי של אוברפיטינג הזיכרון כאשר סט המשימות אינו MeX. הם משתמשים בMAML בשביל לאמן את מטה-מודל שלהם. צריך לציין שעבור בעיות סיווג הם יצרו אפיזודות כך שסט המשימות שלהם לא מקיים את תכונת MeX (למרות ש k-shot, N-way קלאסי הוא כן MeX).

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

לינק למאמר: [מאמר](#)

לינק לקוד: [קוד](#)

נ.ב. אהבתי את החשיבה של מחברי המאמר. מאמר קריא, מוסבר בצורה ברורה

#DeepNightLearners