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:

Teaching with Commentaries

של ג'ף הינטון האגדי ושותפיו.

פינת הסוקר:

המלצת קריאה ממייק: מומלץ לאוהבי מטה-למידה ובעלי רקע בחדו"א2 מתקדם.

בהירות כתיבה: בינונית.

רמת היכרות עם כלים מתמטיים וטכניקות של ML/DL הנדרשים להבנת מאמר: רקע טוב בתחום מטה-למידה, חדו"א ברמה גבוהה.

יישומים פרקטיים אפשריים: ניתן להשתמש בגישה זו למשל לזיהוי דוגמאות המשפיעות ביותר על הביצועים או איתור פאטצ'ים בתמונות מהדאטהסט החשובים למשימה במהלך האימון של הרשת.

פרטי מאמר:

לינק למאמר: זמין להורדה.

לינק לקוד: לא הצלחתי לאתר.

פורסם בתאריך: 5.11.20, בארקיב.

יוצג בכנס: ICLR 2021.

תחומי מאמר:

- שיטות אימון של רשתות נוירונים.
- שיטות מטה למידה (meta-learning) בתחום רשתות הנוירונים.

כלים מתמטיים, מושגים וסימונים:

- משפט הפונקציה הסתומה.
- חישוב נגזרת של פונקציה וקטורית דרך ההופכית של מטריצת הסיאן (hessian).
- קירוב ניומן (neumann) לחישוב הופכית של אופרטור (מטריצה) לינארי.
- רשת לומדת פנימית (inner student network).
- רשת מלמדת (נקראת הרשת המפרשנת במאמר - commentary network).
- אימון פנימי/חיצוני (inner/outer optimization).
- מטה-אימון ,(meta-training).

תמצית מאמר:

כמו שאתם בטח יודעים, למרות הפעילות המחקרית האינטנסיבית בתחום הלמידה העמוקה עדיין קיימות לא מעט סוגיות פתוחות בנושאי אימון, רגולריזציה והבנה של מה שקורה בתוך רשתות נוירונים עמוקות. השאלות האלו נוגעות בסוגיות בסיסיות כמו: איך לאמן רשתות בצורה יותר מהירה, איך להקטין את כמות הדאטה הנדרשת לאימון, איך לשפר את יכולת הכללה רובסטיות של הרשתות.

אחת הגישות המעניינות שהוצעה לאחרונה שמנסה לתת מענה לשאלות אלו נקראת "לומדים ללמד" (learning to teach) המציע לבנות מנגנון חיצוני (רשת בדרך כלל) בשביל לספק לרשת הלומדת תובנות (נקרא לזה גם מטה-מידע בהמשך) לגבי המשימה תוך כדי תהליך האימון. למשל מנגנון כזה יכול לבצע משקול של דוגמאות במטרה לעזור לרשת הלומדת "לרכז את המאמץ" בדוגמאות החשובות. דוגמא אחרת של גישה זו יכולה להיות רשת עזר ה"מייעצת" איך לבנות דוגמאות (למשל ע"י ערבוב של דוגמאות מדאטה סט) הגורמים לרשת הלומדת לבנות ייצוג חזק של דאטה.

מאמר זה מציע מסגרת כללית לגישה זו(הנקראת למידה עם פרשנויות) ומציע תהליך אחיד להפקה של מטה-מידע (פרשנות) עי" רשת חיצונית (מפרשנת) N_com, תוך כדי "הסקת מסקנות" העולות בתהליך האימון של רשת N_st (נקראת הרשת הלומדת) על סט האימון. אז בואו נבין איך כל זה עובד בעצם? נניח שאנו רוצים למצוא איזושהי טרנספורמציה (לדוגמא משקול/ערבוב) של דוגמאות בדאטה סט במטרה לשפר את הביצועים של הרשת הלומדת N_st והמטרה של הרשת המפרשנת N_com הינה למצוא את הטרנספורמציה הזו וזה למעשה מהווה הפלט שלה Out_com. במקרה זה תהליך האימון מכיל את השלבים הבאים:

- אופטימיזציה פנימית: עבור סט משקלים נתון W_com של הרשת N_com, מאמנים את N_st (כמה איטרציות של GD על משקלי W_st). במקרה הזה מפעילים טרנספורמציה Out_com המופקת עי" (N_com, W_com) על הדאטה של סט האימון ומאמנים את N_st עליו. הפלט של השלב הזה הוא המשקלים W_st של N_st.

- אופטימיזציה חיצונית: מחשבים את הלוס של N_st עם סט המשקלים W_st מהשלב הקודם על סט ולידציה שעובר טרנספורמציה הניתנת עי" (N_com, W_com). כאן מאמנים את N_st על משקלי W_com כלומר מבצעים כמה איטרציות של GD אבל הפעם על למשקלי W_com.

- חוזרים על הצעדים אלו T פעמים כאשר T זה מספר האיטרציות של מטה אימון.

(1) Initialize commentary parameters ϕ and student network parameters θ

(2) For M steps:

- (i) Compute the student network's training loss, $\mathcal{L}_T(\theta, \phi)$.
- (ii) Compute the gradient of this loss w.r.t the student parameters θ.
- (iii) Perform a single gradient descent update on the parameters to obtain $\hat{\theta}$ (Note this is implicitly a function of ϕ, i.e. $\hat{\theta}(\phi)$).
- (iv) Compute the student network's validation loss, $\mathcal{L}_V(\phi)$.
- (v) Compute $\frac{\partial \mathcal{L}_V}{\partial \hat{\theta}}$.
- (vi) Approximately compute $\frac{\partial \hat{\theta}}{\partial \phi}$ with equation 4, using a truncated Neumann series with a single term and implicit vector-Jacobian products [17].
- (vii) Compute the overall derivative $\frac{\partial \mathcal{L}_V}{\partial \phi}$ using (v) and (vi), and update ϕ.
- (vii) Set $\theta \leftarrow \hat{\theta}$.

(3) Output: $\hat{\phi}$, the optimized parameters of the commentary.

הסבר של רעיונות בסיסיים:

קודם כל נציין כי הגרדיאנט של משקלי N_com משערך את השינוי בלוס של N_st ביחס לשינוי במשקלים של N_com. אבל צריך לזכור שבשביל לחשב את הלוס של האופטימלי של N_st עבור משקלי N_com נתונים, המשקלים של N_st עוברים כמה איטרציות (אולי די הרבה) של GD במטרה למזער את הלוס שלה. לכן כדי לחשב את הגרדיאנט של הלוס של N_st לפי משקלי W_com צריך "לגלגל את כל האיטרציות על משקלי N_st" ל- W_com שזה יכול להיות די כבד חישובית כאשר N_com הינה רשת גדולה.

גם אם נחליט להשתמש רק באיטרציה אחת בתהליך האופטימיזציה הפנימית (זה מה שעשו במאמר) עדיין של לנו בעיה עם חישוב הגרדיאנט של הלוס לפי W_com. הבעיה הזו נובעת מהעובדה שגרדיאנט זה שווה למכפלה של הגרדיאנט של הלוס לפי W_st (שזה ניתן לחשב אותו בצורה הסטנדרטית של גזירת הלוס של רשתות) והנגזרת של וקטור משקלים W_st לפי לוקטור משקלים W_com . נזכיר ש W_st תלוי ב- W_com בצורה לא מפורשת כי בשלב האופטימיזציה הפנימית W_st מחושב על הדאטה סט אחרי הפעלת עליו טרנספורמציה המוגדרת עי" W_com. נגזרת זו היא בעצם מטריצה (W_com ו- W_st הם וקטורים) שמימדיה עלולים להיות די גבוהים. נניח ש N_st ו- N_com הם רשתות לא גדולות בגודל של מיליון משקלים אז הנגזרת הזו תהיה מטריצה בגודל מיליון על מיליון ותידרש כמות זיכרון עצומה בשביל לאחסן אותה.

לכן המאמר מציע להשתמש במשפט הפונקציה הסתומה עבור הנגזרת של הלוס (של השלב החיצוני L_out) לפי W_st. משפט זה מאפשר לתאר את הנגזרת הבעייתית עי" מכפלה של הופכית הסיאן של לוס של השלב הראשון L_in לפי W_st והמטריצה של הנגזרות המעורבות לפי W_com ו- W_in של L_in. למי שלא זוכר הסיאן זה מטריצה המורכבת מהנגזרות שניות של L_in לפי הזוג של רכיבים של W_com ו- W_st. צריך לזכור שהפירוק לעיל מתקיים בסביבת נקודה שבה הנגזרת של L_in לפי W_stn מתאפסת. העובדה שלא ניתן למצוא אותה במדויק וזה יכול להשפיע בצורה שלילית על התהליך המטה-למידה.

גם אחרי הפירוק הזה יש לנו בעיה - והיא טמונה בחישוב של הופכית של ההסיאן של L_in לפי W_st. אפילו עבור W_st בגודל יחסית לא גדול חישוב ההופכית (ולפעמים ההסיאן עצמו) יכול להיות מאוד כבד וידרוש כמות עצומה של הזיכרון והזמן. בשביל להקל על ההיבט החישובי משתמשים בקירוב נוימן עבור ההופכית מוכפלת בגרדיאנט של L_in לפי W_st (מאמר של לוריין) תוך שימוש בצורת עדכון של GD. דרך אגב נוסחת נוימן מאתרת הופכית של אופרטור לינארי בתור טור אינסופי של החזקות שלה (מוזזות במינוס מטריצה היחידה). במאמר משתמשים רק באיבר אחד של קירוב זה.

הישגי מאמר:

במאמר מראים 3 דרכים להשתמש בגישה זו לשיפור תהליך האימון של רשתות נוירונים:

- חישוב של משקול על דוגמאות מסט האימון עי״ W_com (דוגמאות עם משקל גבוה משפיעים יותר על הלוס). מעניין שהם גם בדקו את הביצועים של השיטה שלהם בתרחיש למידת few-shot שזו משימת מטה-למידה קלאסית. המטרה בלמידת few-shot היא לאמן רשת חיצונית (מטה) על מספר משימות (שנלמדות בפועל עי״ הרשת הפנימית) במטרה ללמד אותה להפיק תכונות משותפות של כל המשימות (שימו לב שזה מקרה פרטי של הפרדיגמה הכללית שהוצעה במאמר). כאשר מגיע משימה חדשה הרשת החיצונות מסוגלת לכייל את את עצמה עם כמות קטנה של דאטה במשימה הזו. MAML זו אחת דרכים לפתור בעיה זו והיא מאמנת רשת חיצונית לאתחול המשקלים של הרשת הפנימית שתאפשר לה להגיע לביצועים טובים על משימה חדשה במספר איטרציות GD קטן. אז הם מראים שהשילוב של MAML עם משקול דוגמאות הנבנה עי״ W_com גורם לשיפור ביצועים משמעותי. המאמר מראה שיפור בביצועים עבור הדאטה סטים MiniImageNet ו- CUB200-2011.

- בנייה של מקדמים ערבוב אופטימליים עבור הדוגמאות (בדומה ל mixup). כאן דוגמא מעורבבת נבנית כסכום קמור של שתי דוגמאות: x_mix = ax_1 + (1 - a)x_2 כאשר מטרת W_com הינה לחשב את מקדמי a האופטימליים לביצועי משימת סיווג(מאמר mixup המקורי מגריל אותם מהתפלגות בטה). מעניין שכאן ״למידה עם פרשנויות״ מנצחת את mixup ב- CIFAR10 ומציגה ביצועים קצת פחות טובים ממנה על CIFAR10 (דאטה סט יותר קטן).

- חישוב מסכות על תמונות לשיפור הפיצ'רים המופקים עי״ רשת. כאן W_com בעצם מחפשת אזורים ״חשובים״ בתמונה שכדאי לרשת הלומדת להתרכז עליהם. המאמר מראה באופן ויזואלי שהמסכות שהוא מפיק אכן מתמקדות באזורים החשובים של תמונות ומראים באופן כמותי את עדיפותן על פני שיטות אחרות לבניית מסכות.

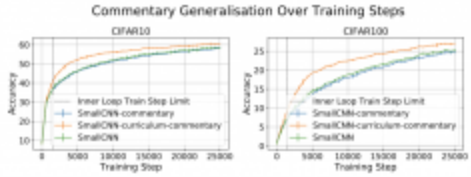

Figure 2: Example weight curricula can speed up training. Test-set accuracy curves on CIFAR10/100 when using curriculum commentaries, non-curriculum commentaries, and no commentaries, during student network training. The learned curriculum commentary network which generates per-iteration example weights results in learning speed improvements. This learning speed improvement holds when the student network is trained for many more steps than the number of inner loop update steps used during commentary network training (1500 steps). This demonstrates that the commentaries generalise to longer training times.

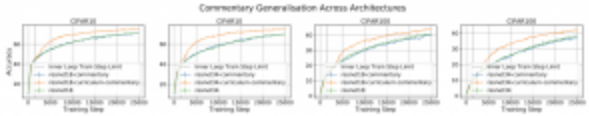

Figure 3: Example weight curricula generalise across network architectures. Using a learned curriculum commentary network trained with a simple 2 block CNN student, we apply these example weights to train two ResNet architectures. This gives improved test set accuracy curves for ResNet students also, indicating that the commentaries generalise across architectures.

נ.ב. מאמר מציע מסגרת כללית לשיפור של תהליך למידה של רשתות נוירונים שניתן להשתמש בה למגוון רחב של משימות של הלמידה העמוקה. הגישה שלהם גם עוזרת להפיק תובנות חדשות תוך כדי תהליך האימון של רשתות. אני מני שעוד נשמע על שימושים רבים של גישה זו...

#deepnightlearners

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

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