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:

Sharpness-Aware Minimization for Efficiently Improving Generalization

פינת הסוקר:

המלצת קריאה ממייק: חובה לאלו שמתעניינים מה קורה מאחורי הקלעים בתהליך אימון של רשתות נוירונים.

בהירות כתיבה: גבוהה מאוד.

רמת היכרות עם כלים מתמטיים וטכניקות של ML/DL הנדרשים להבנת מאמר: היכרות טובה עם שיטות אופטימיזציה עבור בעיות עם משתנים מרובים.

יישומים פרקטיים אפשריים: שיפור יכולת הכללה של רשתות על ידי החלפת בעיית מזעור לוס הרגילה ב-SAM.

פרטי מאמר:

לינק למאמר: זמין להורדה.

לינק לקוד: כאן.

פורסם בתאריך: 04.12.20, בארקיב.

הוצג בכנס:ICLR 2021.

תחום מאמר:

- חקר שיטות אופטימיזציה לאימון של רשתות נוירונים.

כלים מתמטיים, מושגים וסימונים:

- יכולת הכללה של רשת נוירונים.
- Gradient Descent -GD.
- הסיאן (Hessian) של פונקציה.
- בעיית הנורמה הדואלית (dual norm problem).

תמצית מאמר:

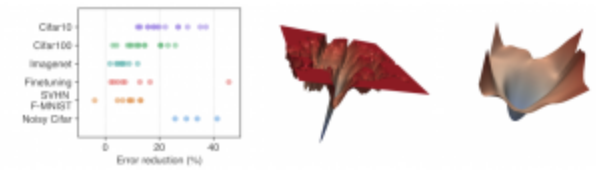


Figure 1: (left) Error rate reduction obtained by switching to SAM. Each point is a different dataset / model / data augmentation. (middle) A sharp minimum to which a ResNet trained with SGD converged. (right) A wide minimum to which the same ResNet trained with SAM converged.

המאמר הנסקר מציע ניסוח חדש לבעיית האופטימיזציה המתרחשת בזמן אימון רשתות נוירונים. במקום מציאת וקטור משקלים, הממזער פונקציית לוס (לסט דוגמאות נתון), המאמר מציע לפתור בעיית אופטימיזציה, שמטרתה למצוא מינימום סביבתי של פונקציית לוס. כלומר, במקום להשתמש ב GD לאיתור המינימום המוחלט, ולעדכן את המשקלים לכיוון מינימום אבסולוטי, האלגוריתם המוצע מכוון לנקודה שבסביבתה פונקציית הלוס תקבל ערכים מינימליים.

בנוסף המאמר מוכיח באופן ריגורוזי כי הפתרון בעיית אופטימיזציה שהם מציעים (הנקרא SAM sharpness aware minimization) תורם באופן חיובי ליכולת הכללה של המודל המאומן.

רעיון בסיסי:

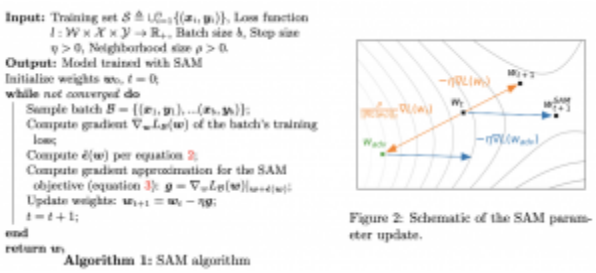
כמו שאתם בטח יודעים הרוב המוחלט של רשתות הנוירונים המודרניות הינן overparameterized בצורה משמעותית. משתמע מכך כי אופטימיזציה של משקלי רשת על סמך ערך של פונקצית לוס בנקודה בלבד (!!) עלול להוביל למודלים בעלי יכולת הכללה נמוכה(קרי overfitting). הסיבה המרכזית לכך הינה מבנה גיאומטרי מאוד מורכב ולא קמור של משטח הלוס. הדוגמא הקלאסית לכך הינה המקרה שבו המינימום של פונקציית לוס "חד" מאוד. כלומר אפילו בסביבתה המאוד קרובה של נקודת המינימום הערכים של פונקציית הלוס הינם גבוהים משמעותית מערכה בנקודת המינימום. נקודה מינימום זו עלולה להיות תוצאה של דאטה רועש ותוביל למודל עם יכולת הכללה נמוכה (overfitting). המאמר מציע פתרון למצב זה עי" ניסוח בעיית אופטימיזציה שמתחשבת לא רק בערך של פונקצית לוס בנקודה, אלא לוקחת בחשבון את ערכי הלוס בסביבתה. כלומר הניסוח המוצע (SAM) לוקח בחשבון גם את התכונות הגיאומטריות של משטח הלוס בסביבות הנקודה באופן מפורש.

תקציר מאמר:

קיימות מספר רב של שיטות המנסות להגדיל את יכולת ההכללה של מודלים בלמידת מכונה. את הפתרונות שהוצעו אפשר לחלק לשתי משפחות עיקריות: הראשונה הינה שינוי האופטימייזר (Momentum, RmsProp, ADAM וכדומה) והשנייה כוללת שינויים בתהליך האימון עצמו (עצירה מוקדמת, BatchNorm, עומק סטוכסטי, אוגמנטציות של דאטה והרבה אחרים). שיטות אלו מנסות לפתור את אותה בעיית אופטימיזציה של מזעור פונקציית לוס בדרכים שונות. לעומתו המאמר הנסקר מציע להחליף את בעיית אופטימיזציה עצמה (!!!).

פרטים טכניים:

פונקציית הלוס המוצעת L מכילה שני איברים - הראשון הוא הלוס המקסימלי בסביבה קטנה של הנקודה w (גודלה של סביבה זו הינו היפר-פרמטר) והשני הינו איבר רגולריזציה סטנדרטי עם נורמת Lp של w (זה דומה לשיטת אופטימיזציה הנקראת proximal point). מעניין כי עבור וקטור משקלים w, ניתן לרשום את Lp כסכום של ההפרש בין הערך המקסימלי של פונקציית לוס בסביבת w (במאמר, הפרש זה נקרא "חדות" - sharpness) ואיבר רגולריזציה חדש שהוא הסכום של נורמת Lp של וקטור המשקלים w וערך הלוס בנקודה w.



Input: Training set $S \triangleq \cup_{i=1}^{n}\{(x_i, y_i)\}$, Loss function $l : \mathcal{W} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$, Batch size b, Step size $\eta > 0$, Neighborhood size $\rho > 0$.
Output: Model trained with SAM
Initialize weights $\mathbf{w}_0$, $t = 0$;
while *not converged* **do**
  Sample batch $B = \{(x_1, y_1), ..., (x_b, y_b)\}$;
  Compute gradient $\nabla_{\mathbf{w}} L_B(\mathbf{w})$ of the batch's training loss;
  Compute $\hat{\epsilon}(\mathbf{w})$ per equation 2;
  Compute gradient approximation for the SAM objective (equation 3): $\mathbf{g} = \nabla_{\mathbf{w}} L_B(\mathbf{w})|_{\mathbf{w}+\hat{\epsilon}(\mathbf{w})}$;
  Update weights: $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \mathbf{g}$;
  $t = t + 1$;
end
return $\mathbf{w}_t$

Algorithm 1: SAM algorithm

Figure 2: Schematic of the SAM parameter update.

ההיבט התיאורטי:

המאמר הנסקר מוכיח כי עבור סט אימון נתון, הלוס של SAM בכל נקודה w מהווה חסם עליון על הלוס על ה- population (שממנה סט האימון נדגם) בהסתברות גבוהה (המשפט שמוכח במאמר טיפה יותר כללי ועובד על משפחה יותר רחבה של פונקציות רגולריזציה). כמובן הכל תחת תנאים טכניים על התפלגות שממנה הדאטהסט נדגם. בעצם המשפט הזה אומר שפתרון בעיית SAM מוביל למודל בעל יכולת הכללה טובה. ההוכחה היא די לא טריוויאלית ומערבת חסמי PAC בייסיאניים (מוכללים).

פתרון בעיית SAM:

קודם כל משתמשים בקירוב טיילור מסדר ראשון, בשביל למצוא את הנקודה בסביבה של w עבורה הלוס הוא מקסימלי. אחר כך, הבעיה בנידון מתורגמת לבעיית הנורמה הדואלית הקלאסית, שיש לה פתרון מפורש e_w. אחרי שמציבים את e_w בביטוי של SAM, מקבלים בעיית אופטימיזציה רגילה (בעיית מזעור עם פונקצית מחיר L(e_w)) שפותרים אותה בדרך הסטנדרטית עם gradient

descent. מכיוון e_w מכיל את הגרדיאנט של הפונקציה הלוס המקורית L, הביטוי עבור הגרדיאנט של L(e_w) מכיל מטריצת הסיאן (hessian) של L. חישוב של הסיאן כאשר ל-w יש מאות מיליונים רכיבים זו משימה מאוד כבדה מבחינת משאבי חישוב וזיכרון. אבל לשמחתנו, בביטוי מופיעה מכפלה של הסיאן בוקטור, שלמעשה מאפשרת לחשב את הערך של הגרדיאנט של L(e_w) ללא חישוב ההסיאן. בסופו של דבר, ניתן להריץ את האלגוריתמים שלהם בדומה ל-GD עם כלי גזירה אוטומטיים כמו TensorFlow או PyTorch.

הישגי מאמר:

המאמר הצליח להראות כי הגישה המוצעת מציגה ביצועים עדיפים על פני שיטות אופטימיזציה שונות ומגוונות (כמו סוגים שונים אוגמנטציה, אופטימייזרים שונים ועוד) על מגוון מאוד רחב של דאטהסטים וארכיטקטורות רשת שונות. בכל השוואה הם פשוט החליפו את האופטימיזציה המקורית ב-SAM והשוו את הביצועים על הטסט סט. בנוסף, המאמר השווה את ביצועי SAM עבור דאטהסטים עם לייבלים רועשים וגם אבחן את השינוי בערכים העצמיים של מטריצת הסיאן עבור הפתרון של בעיית SAM.

		CIFAR-10		CIFAR-100	
Model	Augmentation	SAM	SGD	SAM	SGD
WRN-28-10 (200 epochs)	Basic	$\mathbf{2.7}_{\pm 0.1}$	$3.5_{\pm 0.1}$	$\mathbf{16.5}_{\pm 0.2}$	$18.8_{\pm 0.3}$
WRN-28-10 (200 epochs)	Cutout	$\mathbf{2.3}_{\pm 0.1}$	$2.6_{\pm 0.1}$	$\mathbf{14.9}_{\pm 0.2}$	$16.9_{\pm 0.1}$
WRN-28-10 (200 epochs)	AA	$\mathbf{2.1}_{\pm < 0.1}$	$2.3_{\pm 0.1}$	$\mathbf{13.6}_{\pm 0.2}$	$15.8_{\pm 0.2}$
WRN-28-10 (1800 epochs)	Basic	$\mathbf{2.4}_{\pm 0.1}$	$3.5_{\pm 0.1}$	$\mathbf{16.3}_{\pm 0.2}$	$19.1_{\pm 0.1}$
WRN-28-10 (1800 epochs)	Cutout	$\mathbf{2.1}_{\pm 0.1}$	$2.7_{\pm 0.1}$	$\mathbf{14.0}_{\pm 0.1}$	$17.4_{\pm 0.1}$
WRN-28-10 (1800 epochs)	AA	$\mathbf{1.6}_{\pm 0.1}$	$2.2_{\pm < 0.1}$	$\mathbf{12.8}_{\pm 0.2}$	$16.1_{\pm 0.2}$
Shake-Shake (26 2x96d)	Basic	$\mathbf{2.3}_{\pm < 0.1}$	$2.7_{\pm 0.1}$	$\mathbf{15.1}_{\pm 0.1}$	$17.0_{\pm 0.1}$
Shake-Shake (26 2x96d)	Cutout	$\mathbf{2.0}_{\pm < 0.1}$	$2.3_{\pm 0.1}$	$\mathbf{14.2}_{\pm 0.2}$	$15.7_{\pm 0.3}$
Shake-Shake (26 2x96d)	AA	$\mathbf{1.6}_{\pm < 0.1}$	$1.9_{\pm 0.1}$	$\mathbf{12.8}_{\pm 0.1}$	$14.1_{\pm 0.2}$
PyramidNet	Basic	$\mathbf{2.7}_{\pm 0.1}$	$4.0_{\pm 0.1}$	$\mathbf{14.6}_{\pm 0.4}$	$19.7_{\pm 0.3}$
PyramidNet	Cutout	$\mathbf{1.9}_{\pm 0.1}$	$2.5_{\pm 0.1}$	$\mathbf{12.6}_{\pm 0.2}$	$16.4_{\pm 0.1}$
PyramidNet	AA	$\mathbf{1.6}_{\pm 0.1}$	$1.9_{\pm 0.1}$	$\mathbf{11.6}_{\pm 0.1}$	$14.6_{\pm 0.1}$
PyramidNet+ShakeDrop	Basic	$\mathbf{2.1}_{\pm 0.1}$	$2.5_{\pm 0.1}$	$\mathbf{13.3}_{\pm 0.2}$	$14.5_{\pm 0.1}$
PyramidNet+ShakeDrop	Cutout	$\mathbf{1.6}_{\pm < 0.1}$	$1.9_{\pm 0.1}$	$\mathbf{11.3}_{\pm 0.1}$	$11.8_{\pm 0.2}$
PyramidNet+ShakeDrop	AA	$\mathbf{1.4}_{\pm < 0.1}$	$1.6_{\pm < 0.1}$	$\mathbf{10.3}_{\pm 0.1}$	$10.6_{\pm 0.1}$

Table 1: Results for SAM on state-of-the-art models on CIFAR-{10, 100} (WRN = WideResNet; AA = AutoAugment; SGD is the standard non-SAM procedure used to train these models).

לייבלים רועשים:

SAM הציג שיפור ניכר כאשר הוא מופעל באימון על דאטהסטים עם לייבלים רועשים. בעצם זה לא מפתיע, כי החוזק העיקרי של האלגוריתם הוא מניעת התכנסות למינימום "חד", ונוכחות לייבלים רועשים בכמות ניכרת עלול להוביל בקלות למינימומים כאלו באלגוריתמים אופטימיזציה קלאסיים.

מבנה ההסיאן בסביבת נקודת אופטימום:

בשביל לאשש את ההנחות לגבי היכולות של SAM במניעת המינימומים החדים, המאמר בחן את הערכים העצמיים (ע"ע המקסימלי ובנוסף גם היחס בין ע"ע המקסימלי לבין כמה ע"ע הגבוהים ביותר חוץ מהמקסימלי) של ההסיאן בנקודות אופטימום שנמצאו ע"י SAM מול אלו שנמצאו באמצעות אלגוריתמים אחרים. הרי ידוע שככל שהמינימום יותר חד, יש להסיאן גם ערכים עצמיים גבוהים יותר וגם היחס בין ע"ע המקסימלי לבין ע"ע-ם הגבוהים ביותר, חוץ מהמקסימלי, גבוה יותר גם כן. המאמר הראה כי שימוש ב- SAM מוריד את שני מדדים אלו בצורה מאוד משמעותית.

דאטהסטים:

CIFAR10, CIFAR100, Flowers, Stanford_cars, Birdsnap, Food101, Oxford_IIIT_Pets, FGVC_Aircraft, Fashion-MNIST וכמה אחרים.

ארכיטקטורות רשת שנבחנו:

Wide-ResNet-28-10, Shake-Shake , EffNet, TBMSL-Net, Gpipe וכמה אחרים.

נ.ב.

מאמר מאוד חשוב המציע שיטה מאוד מעניינת לשיפור יכולת הכללה של רשתות. לדעתי, יש לשיטה פוטנציאל רציני להיכנס לארגז כלים סטנדרטי לאימון רשתות. התרשמתי גם המשוואות הרבות והמגוונות מול שיטות אחרות שנעשו במאמר.

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.