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:

Supermasks in Superposition

פינת הסוקר:

המלצת קריאה ממייק: מומלץ מאוד - יש במאמר שני רעיונות מגניבים.

בהירות כתיבה: בינונית פלוס.

רמת היכרות עם כלים מתמטיים וטכניקות של ML/DL הנדרשים להבנת מאמר: הבנה בסיסית בתחום למידה מתמשכת (continual learning), בלמידה מתמשכת וברשתות הופפילד.

יישומים פרקטיים אפשריים: בניית רשת נוירונים גדולה עם משקלים קבועים המשמשת לביצוע משימות מרובות (דומות באופי).

פרטי מאמר:

לינק למאמר: זמין להורדה.

לינק לקוד: זמין כאן.

פורסם בתאריך: 22.10.20, בארקיב.

הוצג בכנס: NeurIPS 2020.

תחומי מאמר:

- שיטות למידה מתמשכת (continual learning) עם רשתות נוירונים.
- למידת משימות מרובות (multi-task learning) עם רשתות נוירונים.

כלים מתמטיים, מושגים וסימונים:

- מסכות בינאריות על משקלים ברשתות נוירונים.
- שכחה קטסטרופלית ברשתות נוירונים
- רשתות הופפילד (HN).
- אנטרופיה (זה המושג המרכזי שעליו המאמר בנוי).

תמצית מאמר:

המאמר מציע שיטה אימון SupSup של רשת נוירונים גדולה (נקרא לה רשת בסיס), המקנה לה יכולת לבצע כמה משימות שונות. המשקלים של רשת הבסיס הינם קבועים לכל המשימות. בעצם לוקחים רשת, מאתחלים את משקליה באופן רנדומלי ומשתמשים באותה רשת לחיזוי עבור משימות שונות. הדרך לבצע זאת זה ללמוד סט של מסכות בינאריות נפרד (0 או 1) לכל משימה ועבור כל משימה "להלביש את סט המסכות" שלה על רשת הבסיס בזמן אינפרנס. בעצם מסכה בינארית כזו מדליקה או מכבה קשרים בין נוירונים שונים ברשת. בדרך זו מתגברים על תופעת השכחה הקטסטרופלית שעלולה להתרחש אם מאמנים (מכיילים) רשת עבור משימה חדשה.

Figure 1: **(left)** During training SupSup learns a separate supermask (subnetwork) for each task. **(right)** At inference time, SupSup can infer task identity by superimposing all supermasks, each weighted by an α_i, and using gradients to maximize confidence.

המאמר מגדיר 4 תרחישים טיפוסיים של למידה מתמשכת ומציע שיטה לאמן רשת בסיס אחת לביצוע של משימות מרובות עבור כל אחד מהם:

- המשימות ידועות (מזוהות) גם במהלך האימון וגם במהלך האינפרנס (כלומר כל פעם שאנו מקבלים משימה אני יודעים איזו משימה זו): תרחיש זה מסומן כ- GG.
- המשימות ידועות במהלך האימון ולא ידועות במהלך האינפרנס (עם לייבלים משותפים). קרי אנחנו צריכים לנחש איזו משימה קיבלנו באינפרנס בשביל להבין לאיזו משימה באימון היא שייכת (איזו מסכה לבחור): מסומן עי" GN.
- המשימות ידועות במהלך האימון ולא ידועות במהלך האינפרנס עם לייבלים שונים. כלומר שכבת היציאה של הרשת צריכה להיות בגודל סכום של מספר הלייבלים עבור כל המשימות: GNu.
- המשימות לא יודעות לא בזמן האימון ולא בזמן האינפרנס (הלייבלים חייבים להיות משותפים כאן). כאן באימון אנו מקבלים משימה ולא יודעים איזו משימה קיבלנו. כלומר כל פעם אנו צריכים להחליט האם להשתמש במסכה הקיימת או לאמן מסכה חדשה. דבר דומה קורה גם באניפרנס: NN.

רעיונות בסיסיים:

יש כמה רעיונות מעניינים במאמר. ניתן להפריד אותם בגדול לשני סוגים:

סוג 1: שיטות לפתרון לכל התרחישים, המתוארים מעלה, של בעיות הלמידה המתמשכת:

- **משימות GG:** לבחור מסכה המתאימה עבור משימת אינפרנס.
- **משימות GN ו- GNu:** המאמר מציע לתאר את המסכה עבור משימת אינפרנס כצירוף לינארי של כל המסכות שאומנו ולחפש את המקדמים ע"י מציאת המקדם שהעלייה בו מביאה לירידה הכי גדולה באנטרופיה של פלט הרשת עבור המשימה. כלומר המקדם, שהגרדיאנט של האנטרופיה השלילית (מוכפלת ב- -1) של רשת הבסיס לפיו, הינה מקסימלית. זה מאפשר לא להריץ את הרשת עבור כל מסכות בזמן אינפרנס (אם יש מאות רבות או אלפי מסכות ודאטה סטים גדולים, זה יכול להיות משמעתי). במקום זאת מריצים את המשימה ברשת פעם אחת עם ממוצע של כל המשקלים (הם קוראים לזה One-Shot) ומחשבים את הגרדיאנט. נציין ש- One-Shot מבוסס על גרדיאנט אחד בלבד של אנטרופית רשת הבסיס שהיא פונקציה לא קמורה ביחס למקדמים של המסכות. עובדה זו עלולה להביא לבחירה של מסכה לא נכונה. בשביל להתגבר על בעיה זו המחברים הציעו לבחור את המקדם בתהליך איטרטיבי. כל פעם מאפסים חצי מהמקדמים עם גרדיאנטים הכי נמוכים עד שנשארים עם מסכה אחת.
- **משימות NN:** עושים משהו דומה למתואר בסעיף הקודם אבל אם לא נמצא מקדם שהשינוי בו מביא לעלייה משמעותית באנטרופיה (מחשבים softmax על כל הגרדיאנטים), מאמנים מסכה חדשה. אחרת לוקחים את המקדם עבורו התקבל המקסימום ומשתמשים במסכה שלו. דרך אגב הם לא ציינו אופציה פשוטה נוספת לפתרון: במקום לאמן מסכה חדשה (במקרה שצריך) אפשר לאמן מקדמים של הקומבינציה הלינארית הכי אופטימליים מבחינת הביצועים - הרי צריך פחות מקום בשביל לאחסן את המקדמים מאשר המסכה.

בנוסף המחברים מציעים איזה טריק נחמד התורם משמעתית לשיפור בביצועים. הם מוסיפים "לייבלים מלאכותיים" למשימה כלומר נוירונים נוספים לשכבה האחרונה של רשת הבסיס. למשל אם אנו מאמנים מסווג MNIST עם 10 קלאסים אז השכבה האחרונה תכלול, נגיד, 100 נוירונים כאשר 90 הנוירונים שהוספו ל- 10 הרגילים שייכים לקטגוריות "לא קיימות". המאמר מנצל את ערכי הנוירונים האלו בשביל להחליף את מדד הנגזרת לפי המקדמים של צירוף לינארי של המסכות.

Table 1: Overview of different Continual Learning scenarios. We suggest scenario names that provide an intuitive understanding of the variations in training, inference, and evaluation, while allowing a full coverage of the scenarios previously defined in [49] and [55]. See text for more complete description.

Scenario	Description	Task space discrete or continuous?	Example methods / task names used
GG	Task Given during train and Given during inference	Either	PNN [42], Batchil [51], PSP [4], "Task learning" [55], "Task-IL" [49]
GNs	Task Given during train, Not inference; shared labels	Either	IFWC [23], IL1 [54], "Domain learning" [55], "Domain-IL" [49]
GNu	Task Given during train, Not inference; unshared labels	Discrete only	"Class learning" [55], "Class-IL" [49]
NNs	Task Not given during train Not inference; shared labels	Either	BGD, "Continuous/discrete task agnostic learning" [55]

סוג 2: שיטות לשמירה יעילה ולהפקת מסכה מתאימה למשימה (עבור משימות GN)

המאמר מציע לשמור מסכות ברשת הופפילד (HN) כאשר שומרים את כל המסכות בתוכה עי" ביצוע עדכון המשקלים שלה (של HN). בזמן האינפרנס מנסים לאתר את המסכה האופטימלית עי" חיפוש מינימום של הסכום הממשוקל של פונקצית האנרגיה של HN (הלוס הרגיל שלה) והאנטרופיה של המסכה ברשת הבסיס (עבור המשימה שבנידון).

פינת האינטואיציה: עכשיו ניקח כל רעיון המוצע במאמר וננסה להבין את הרציונל מאחריו.

<u>גרדיאנט של אנטרופיית שכבת פלט של רשת בסיס ביחס מקדמים של קומבינציה לינארית של מסכות:</u>

קודם כל ככל שהרשת יותר בטוחה בחיזוי שלה עבור דוגמא נתונה, האנטרופיה של שכבת הפלט שלה תלך ותרד. למשל לווקטור פלט [0.05, 0.9, 0.05] (הרשת "ממש בטוחה" בחיזוי שלה) יש אנטרופיה נמוכה הרבה יותר מוקטור [0.45, 0.3, 0.3] (הרשת לא "בטוחה"). אז אם גרדיאנט של אנטרופית שלילית של הרשת הוא גבוה עליה במקדם זה תוביל לעלייה באנטרופיה השלילית כלומר לירידה באנטרופיה. ההנחה כאן שאם הרשת פולטת חיזויים "בטוחים" עבור מסכה מסוימת אז כנראה שקיבלנו משימה דומה לזו שהמסכה הזו אומנה. שימו לב שהאנטרופיה מחושבת על כל סט האימון של המשימה שהופך את ההנחה הזו לסבירה.

<u>הוספה של לייבלים מלאכותיים לשכבת הפלט:</u>

זה רעיון שמאוד אהבתי - כאשר אנו מאמנים רשת עם יותר נוירונים בשכבת הפלט עבור משימה מסוימת הרשת לומדת לשים של ערכים שליליים גבוהים בערך המוחלט (ההופכים לאפס עם softmax לאחר מכן). אז אם באינפרנס אנחנו מקבלים שם ערכים שהם לא שליליים גבוהים זה סימן שמשימה זו אינה תואמת למשימה עליה מסכה זו אומנה. אז במקום להשתמש באנטרופיה המאמר מציע לחשב את הלוגריתם של סכום האקספוננטים של הערכים בנוירונים המלאכותיים של השכבה האחרונה. אם יוצא ערך גבוה אז המסכה לא מתאימה למשימה. עם כל היופי ברעיון הזה יש לי תחושה שניתן להשיג תוצאה דומה דרך הכנסת טמפרטורה בסיגמואיד למשחק.

<u>שמירה של מסכות ברשת הופפילד:</u>

בשביל לחסוך במקום אחסון של המסכות המאמר מציע לשמור אותם ברשת הופפילד HN. HN זה בעצם מטריצה שמטרתה לאחסן וקטורים המורכבים מ {-1,1} בצורה חסינה נגד רעש. המסכות של הרשתות מורכבות מ- {0,1} אז עושים טרנספורמציה פשוטה בשביל להפוך אותם לפורמט של HN. כל פעם שרוצים לאחסן וקטור נוסף ב HN מעדכנים את המטריצה שלה עם הווקטור הזה (יש כמה דרכים לעשות זאת - הם השתמשו בכלל עדכון של סטורקי). אז איך קוראים ממטריצת הזכרון הזה? נגיד קיבלנו גרסה מורעשת של ווקטור, מזינים אותו לפונקציות אנרגיה המוגדרות עי" מטריצה זו (בגדול זה צורה ריבועית שלה) ומנסים להביא אותה למינימום. ניתן להוכיח המינימום מתקבל בנקודה השמורה הכי קרובה לקלט המורעש.

אבל במקרה שלנו (זה עובד רק בתרחיש GN) אנחנו לא רק צריכים לאתר את הוקטור השמור הקרוב ביותר לקלט(שהוא תמיד ממוצע של כל המסכות) אלא למצוא מסכה שמביאה למינימום את האנטרופיה של פלט הרשת. אז הם הוסיפו ללוס הרגיל של HN איבר המכיל אנטרופיה של הרשת. בעצם הלוס הינו צירוף לינארי של לוס HN רגיל עם מקדם העולה עם האיטרציות והלוס של אנטרופיה היורד עם מספר האיטרציות. האינטואיציה כאן שבהתחלה זזים בכיוון של המסכה הנכונה וכאשר אנחנו באיזור מבצעים תהליך רגיל של מינימיזציה אנרגיה של HN.

ולסיום אני רוצה לציין כי המאמר משתמש ברשת בסיס די גדולה ומאוד overparameterized. זה מאפשר למצוא מסכות, המאפסות הרבה מהמשקלים שלה, שניתן לאמן אותן לביצוע של משימות שונות. דרך אגב הם לא ציינו איך מתבצע האימון של כל מסכה פר משימה (יש מגוון שיטות).

הישגי מאמר:

המאמר מוכיח את העליונות של השיטה שלהם בכל התרחישים המתוארים מעלה וגם מראים שהביצועים שלהם לא רחוקים מהביצועים האופטימליים של רשת הבסיס למשימה (כאשר הרשת מאומנת לכל משימה בנפרד מחדש). הם גם מראים חיסכון משמעותי במקום איחסון לשיטות עם ביצועים דומים. בנוסף הם גם מראים שהגישה שלהם מאפשרת לאמן אלפי משימות על רשת בסיס אחת עם ביצועים הלא נופלים בהרבה מהביצועים המקסימליים.

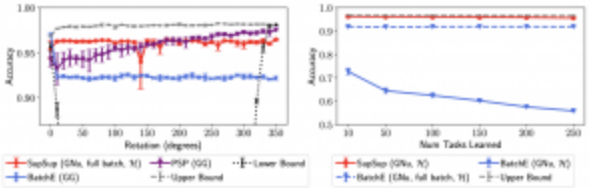

Figure 5: **(left)** Testing the FC 1024-1024 model on RotatedMNIST. SupSup uses **Binary** to infer task identity with a full batch as tasks are similar (differing by only 10 degrees). **(right)** The **One-Shot** algorithm can be used to infer task identity for BatchE [51]. Experiment conducted with FC 1024-1024 on PermutedMNIST using an output size of 500, shown as mean and stddev over 3 runs.

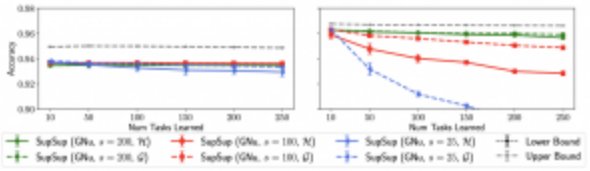

Figure 6: The effect of output size s on SupSup performance using the **One-Shot** algorithm. Results shown for PermutedMNIST with LeNet 300-100 **(left)** and FC 1024-1024 **(right)**.

Algorithm	Avg Top 1 Accuracy (%)	Bytes
Upper Bound	92.55	10222.81M
SupSup (GG)	89.58	195.18M
	88.68	100.98M
	86.37	65.50M
BatchE (GG)	81.50	124.99M
Single Model	-	102.23M

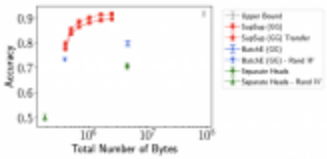

Figure 2: **(left) SplitImagenet** performance in Scenario GG. SupSup approaches upper bound performance with significantly fewer bytes. **(right) SplitCIFAR100** performance in Scenario GG shown as mean and standard deviation over 5 seed and splits. SupSup outperforms similar size baselines and benefits from *transfer*.

דאטהסטים:

GG: SplitCIFAR100, SplitImageNet.

GN: PermutedMNIST, RotatedMNIST, SplitMNIST.

NN: PermutedMNIST.

נ.ב.

המאמר מציע שיטה מבריקה לאימון רשת אחת לביצוע של מספר גדול של משימות. עם זאת צריך לזכור כמה דברים:

1. המשימות שהם אימנו הם באותה דרגת קושי (אני לא בטוח שהעסק יעבוד חלק אם המשימות היו בדרגות קושי שונות - אולי אז צריך לשחק עם מספר האחדים לכל מסכה בנפרד או משהו כזה).
2. המשימות שאומנו הן דומות מבחינה סמנטית. הם לא ניסו לשלב דאטה סטים מדומיינים שונים.
3. המשימות שהם אימנו עליהם הן לא קשות ונשאלת השאלה אולי מבחינת מקום אחסון עדיף לאמן רשת קטנה לכל משימה?

#deepnightlearners

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

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.