

דו"ח עבודת סוף קורס - יישומים ברשתות נוירונים עמוקות (10544)

אביתר גולן, 203311733

אליעזר רווח, 313368102

מרצה: ד"ר שפנייר אסף

תוכן

1	תקציר
1	מבוא
2	עבודות קודמות
3	נתונים
3	שיטות
3	תוצאות הניסויים
4	בעיות שנתקלנו בהן
5	מסקנה

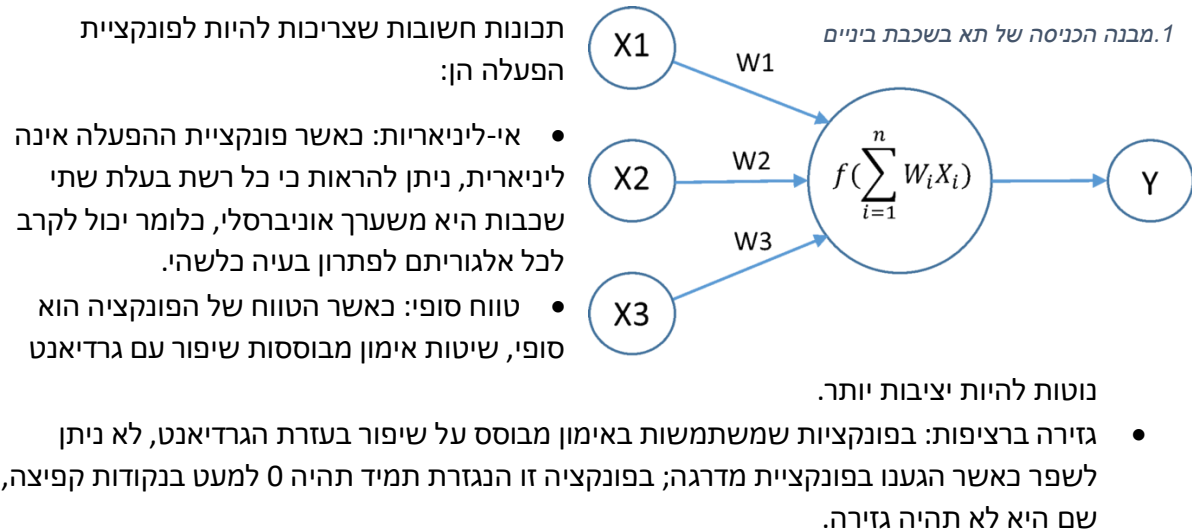
תקציר

בפרויקט זה החלטנו לבדוק האם קיימת פונקציית הפעלה אשר מתאימה את עצמה לנתונים שהרשת מקבלת, ולמבנה הרשת. הפונקציה מקבלת מספר משתנים ומתאימה את עצמה לנתונים בצורה שתוסבר בהמשך הדו"ח. בדקנו אפשרות לפונקציה אחת שתוכל לשמש כפונקציה אוניברסלית לכלל השכבות ואפשרות לפונקציית הפעלה ייחודית לכל שכבה. בנוסף בדקנו אפשרות לקיצור זמן הריצה של הרשת עם הפונקציה שלנו, לה נקרא בקיצור Dynamic, בעזרת המרת הפונקציה מפונקציה אי ליניארית לפונקציה ליניארית למקוטעין, כלומר למערכת של אי-שוויוניים. בתחומים ובמצבים רבים קיבלנו תוצאות מרשימות, עליהן ניתן לקרוא יותר בפרק המתאים ובנספחים.

מבוא

רשתות נוירונים בנויות ממספר שכבות; שכבת כניסה, שכבת יציאה ושכבות ביניים שיושבות ביניהן, כאשר לכל שכבה יש מספר רב של 'תאים', מעין כניסות ייחודיות בשכבה. כל תא בכל השכבות למעט התאים בשכבת הכניסה יכול לקבל קלט ממספר מקורות, וכל תא בכל השכבות למעט התאים בשכבת היציאה יכול להוציא את הפלט שלו למספר תאים בשכבה הבאה. אל שכבת הכניסה, הידועה גם בעגה בשם 'שכבת האינפוטים', נכנס הקלט לרשת אותו אנו רוצים ללמוד. מספר הכניסות בשכבת הכניסה זהה למספר המאפיינים עליהם אנו רוצים ללמוד. משכבת היציאה, קרי 'שכבת האאוטפוטים', יוצא הפלט למידע עליו אנו נלמד. המידע היוצא לרוב מנורמל בשכבה לפני, בעזרת פונקציית Softmax, ע"מ להפוך את הנתונים להתפלגות הניתנת להבנה באופן אינטואיטיבי. מספר התאים בשכבת היציאה זהה למספר השכבות מתוכן נרצה לבחור את המידע שלנו. בעוד שכבות ביניים ('שכבות נסתרות') אינן מרכיב נחוץ לבניית רשת נוירונים, רשת שלא תכיל אותן (ידועה לרוב בשם 'פרספטרון') תהיה מאד מוגבלת בסוג וכמות החישובים שתוכל לעשות. שכבות ביניים מכילות לפחות כמות תאים זהה לזו שבשכבת הכניסה, ולעיתים יותר. מספר הכניסות לכל תא יכול להיות לכל היותר כמות התאים שבשכבה הקודמת (במידה וכל התאים ששכבה הקודמת מחוברים לתא זה, תהליך לו אנו קוראים Fully Connected) בסוף כל שכבת ביניים אנו סוכמים את כל הערכים הנכנסים לתא בשכבה הבאה כפול משקלים נתונים לחיבורים, מוסיף משקל מופרד הנקרא

bias, מועבר את כל הסכום דרך פונקציה הנקראת פונקציית הפעלה (או פונקציית אקטיבציה) המוציאה את הפלט שלה לתא אחד בשכבה הבאה ברשת.



במסגרת הקורס למדנו על מספר סוגי רשתות ועל החשיבות של פונקציות הפעלה לא ליניאריות ברשתות אלו. לקראת אמצע הקורס עלתה לנו המחשבה "האם ניתן לייצר שיטה לבניית פונקציית הפעלה שתתאים עצמה לבעיות שונות, כגון עבודה עם מודלים ונתונים שונים?". לאחר בדיקה עם אסף גילינו כי ניתן לשפר את הציונים בעזרת התאמת הנתונים באיזור שבו הם קרובים לערכי 0, כלומר עבור $\sum (x_i w_i + b_i) \approx 0$ ניתן לשפר את רמת הדיוק באימון בעזרת שימוש בפונקציות המוספות ל-ReLU בתחום הערכים הגדולים מ-0. ההגיון שעומד מאחורי הרעיון הוא הצורך לשפר סביבה של ערכים שדורשים יחס שונה בנגזרת. לדוגמא, עבור ערכים שקרובים ל-0 עדיף שהנגזרת תהיה קטנה יותר בשביל שההתקרבות ל-0 תהיה איטית יותר.

עבודות קודמות

שתי עבודות שאליהן ניתן להשוות את הפרויקט שלנו הן:

- Research on Convolutional Neural Network Based on Improved ReLU Piecewise Activation Function [Guifang Wei, 2018]
- Dynamic ReLU (Yinpeng, et al., 2020)

להלן תמצות פרטי העבודות, והשוני ביניהן לעבודה שלנו;

Yinpeng, et al., 2020	Guifang Wei, 2018	
לשפר את פונקציית ReLU על ידי שינוי השיפועים של חלקי הפונקציה השונים (גם בתחום שקטן מ-0)	למצוא פונקציית הפעלה למקוטעין שתוכל לשפר את התוצאות עבור ערכי X קטנים מאפס על פני ReLU	מטרת העבודה
ImageNet	CIFAR-10	נתונים
פונקציית ההפעלה החדשה הראתה שיפור של בין 3.5% ל-4% ברמת הדיוק על פני ReLU	פונקציית ההפעלה החדשה הראתה שיפור של כ-1.13% ברמת הדיוק על פני ReLU	תוצאות העבודה
אנחנו לא נוגעים בתחום הערכים השליליים, וגם Dynamic אינה	אנו עובדים על תחום הערכים החיוביים הקרובים	איך שונה משלנו?

לאפס בלבד, בעוד הם עבדו על כל תחום השליליים	ליניארית בכל חלקיה השונים (בצורתה המקורית)
--	---

נתונים

בפרויקט זה השתמשנו במאגרי הנתונים CIFAR-10/100 (לגבי CIFAR-10 ניתן לקרוא בנספחים), מאגרי תמונות המכיל כ-60,000 תמונות המחולקות ל-100 מחלקות שונות, כאשר לכל מחלקה יש 500 תמונות לאימון ו-100 תמונות לבחינת איכות הלמידה. המאגר הוא תת-מאגר של מאגר ידוע אשר הכיל מעל 79 מיליון תמונות שנמשכו ממנועי חיפוש בשנת 2008 והוצמדו להן תוויות באופן אוטומטי. המאגר יצא משימוש בשנת 2020. כאשר המאגר CIFAR-100 נוצר החליטו יוצריו לכתוב לו תוויות משלו. מתוך 60,000 תמונות שנמצאות במאגר אנחנו בודקים את מציאת הפרמטרים הטובים ביותר על מקבצים קבועים של בין 500 ל-5,000 תמונות, כלומר למידה על הפרמטרים הטובים ביותר לפי 500 תמונות ולפי עד 5,000 תמונות.

שיטות

$$f(x) = \begin{cases} 0, & x < 0 \\ x + \frac{L}{Q(x+I)^P + Z} - \frac{L}{Q(x-I)^P + Z}, & x \geq 0 \end{cases}$$

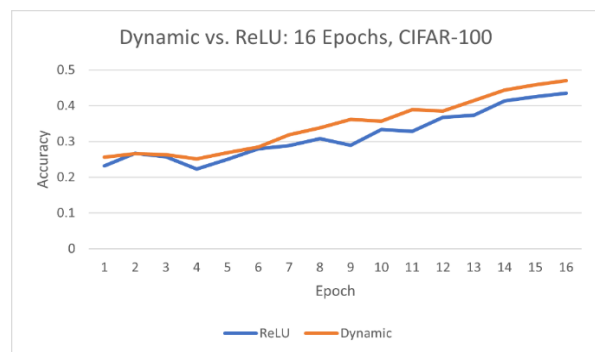
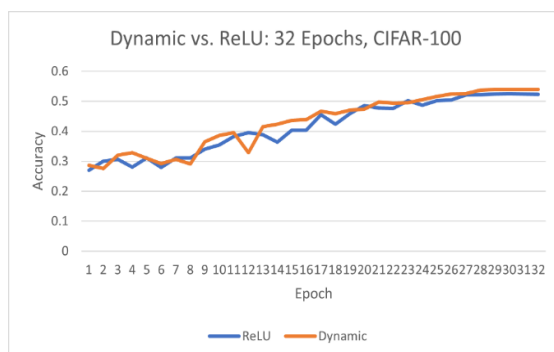
הפונקציה שמצאנו שעובדת טוב על התחום הנ"ל היא

כאשר כל סימני האותיות למעט x הם קבועים הנקבעים לפי למידה חוזרת בלולאות על חלק קטן מהנתונים. החלטנו לבדוק במהלך הפרויקט שני דברים שלדעתנו חשובים: האם עבור כמות זמן נתונה, אפשרי רק ליצור פונקציית הפעלה אחת לכל השכבות, או שניתן ליצור פונקציה ייחודית לכל שכבה שתיתן תוצאות יותר טובות? בנוסף, האם ניתן לשפר את זמן הריצה על פני הרצה מלאה של כל הלולאות לטובת מציאת הפרמטרים הטובים ביותר?

תוצאות הניסויים

בדיקה ראשונית בניסוי הייתה לבדוק האם קיימת פונקציית הפעלה אחידה לכל השכבות (יותר טובה מ-ReLU) על CIFAR-10 (תוצאות בנספחים) ועל CIFAR-100. הניסוי השתמש ברשת נוירונים מסוג ResNet-9. עבור כלל הניסויים שביצענו P ו-Z היו קבועים בערכים 2 ו-10 בהתאמה. תוצאות הניסוי היו להלן:

הניסוי הניב תוצאות מוצלחות עבור מציאת הפרמטרים הטובים ביותר, כאשר הרצת הבדיקות למציאת פרמטרים טובים ביותר עם פחות עידינים הובילה למציאת ערכים טובים יותר. כאשר הרצנו את הפונקציה עם הפרמטרים שמצאנו ראינו שיפור ניכר לעומת ReLU.



2

1

גרפים 1, 2 מציגים את רמת הדיוק של Dynamic לעומת ReLU כפונקציה של מס' העידינים שהאלגוריתם ניצל כדי לאמן את הרשת, בשני הרצות נפרדות. ב-Error! Reference source not found. נעשתה הלמידה על כ-5,000 תמונות, וב-Error! Reference source not found. על כ-25,000 תמונות.

לאחר ניסוי הנ"ל, ביצענו ניסוי נפרד שבו נבדקה האפשרות לפונקציית הפעלה מותאמת אישית לכל שכבה ברשת, כלומר עבור כל שכבה ניצור פונקציית הפעלה שונה עם סט פרמטרים משלה. הניסוי בוצע בעזרת נתוני CIFAR-100 ו-ResNet-9. תוצאות הניסוי היו לא טובות; מציאת הפרמטרים הטובים בעזרת כ-1000 תמונות מכלל סט האימון לקחה מעל יומיים, ועבור הרצה עם 16 עידינים שאורכה היה ארוך מ-ReLU בערך

ב-30% עם הפרמטרים הנ"ל, לא קיבלנו תוצאות יותר טובות מ-ReLU. בעקבות זמן ריצה ארוך למדי של אלגוריתם חיפוש הפרמטרים, לא התאפשר לנו לחפש פרמטרים יותר לעומק או להיעזר בכמות נתונים גדולה יותר.

בעיות שנתקלנו בהן

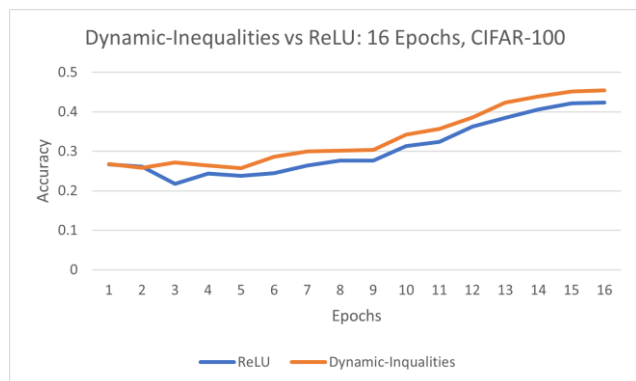
הבעיה הראשונה שצצה הייתה בדיקת האלגוריתם על גבי ResNet-18; זמני הריצה של מציאת הפרמטרים הטובים ביותר, בין אם לפונקציה גלובלית או לפי פונקציות מותאמות לשכבות, הייתה איטית מדי כדי למדוד ונעצרה ע"י התנתקויות ממערכות החישוב של Google Colab שבהן השתמשנו בגלל מכסת זמן המחשוב אותה עברנו. החלטנו לבסוף להוריד ל-ResNet של 9 שכבות.

הבעיה השנייה שהופיעה במהלך הרצת התוכנית למציאת הפרמטרים הטובים ביותר הבנו כי זמן הריצה מתארך משמעותית עקב הזמן שלוקח למצוא נגזרת לפונקציה שבנינו. הפתרון שאליו הגענו היה לנסות לקרב לפונקציה בעזרת סדרה של משוואות ישר הכלואות בין תחומים של נקודות משמעותיות.

האלגוריתם שבו השתמשנו ע"מ למצוא את מערכת האי-שוויוניים עבור פונקציה f היא:

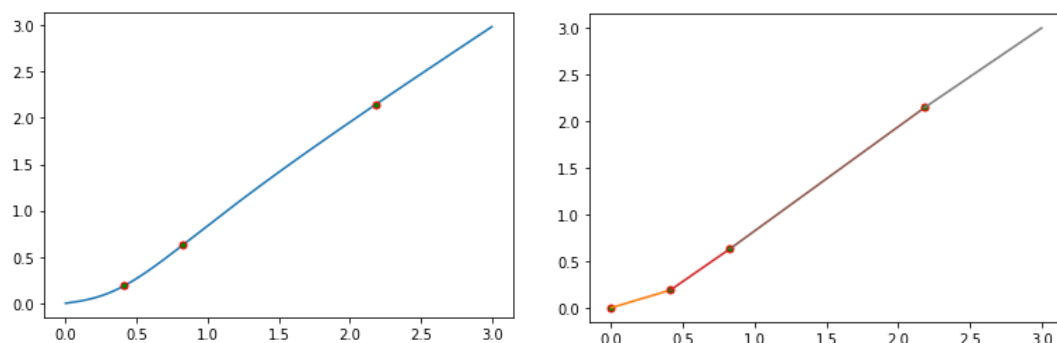
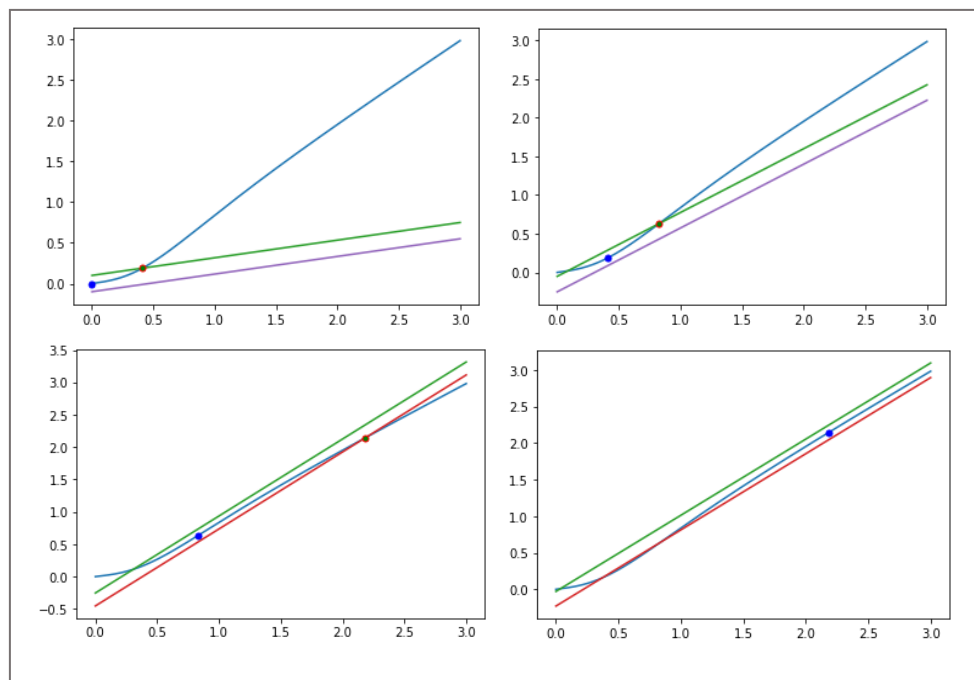
1. קובעים ערך k שרירותי, וערך סף max לתחום ה- x .
2. קובעים ערך $i = 0, x_0 = 0$.
3. עבור הנקודה x_i , נמצא את ערך השיפוע m של הפונקציה f בנקודה.
4. נייצר שני ישרים מקבילים אותם נגדיר הבא $y = m * x_i \pm k + b$.
5. נמצא את נקודת החיתוך Q הקרובה ביותר מימין של אחד משני הישרים עם הפונקציה f .
6. נקבע $x_{i+1} = Q$, נגדיל את i באחד.
7. נבדוק האם $x_i > max$, אם לא נחזור לשלב 3, אם כן נסיים את הריצה ונחזיר את נקודות החיתוך.
8. נייצר קווים ישרים שעוברים בין נקודות החיתוך.
9. הנגזרות של ישרים הם תמיד קו ישר (כלומר ערך קבוע).

ככל שערך ה- k קרוב יותר ל-0 נשאף לפונקציה f , וככל שיתקרב ל- ∞ הישר ישאף ל-ReLU. הסיבה שאנו שמים שני ישרים מקבילים במרחק k היא כדי שישרים של האי-שוויוניים לא התרחקו במרחק גדול מ- k מהפונקציה המקורית. הצלחנו לשפר את התוצאות בעזרת האי-שוויוניים, אם כי זמן הריצה נהיה ארוך למדי – פי ארבע מזמן הריצה הרגיל של Dynamic – עניין הדורש בדיקה ברמת הקוד מדוע לוקח לו יותר זמן לחשב את אי-השוויוניים.



הרצה של האלגוריתם למציאת מערכת אי-שוויוניים על פונקציית Dynamic שנמצא בשלב יותר מוקדם והרצה להשוואה מול ReLU.

ערך ה- k היה על 0.1 ומצאנו ארבעה אי-שוויוניים



מסגרת עליונה: שלבים 3-7 של האלגוריתם, כאשר העיגול הכחול מסמן את הנקודה x_i שמהשיפוע שלו מייצרים את שני הישרים המקבילים, והעיגול השני את נקודת החיתוך הקרובה ביותר של הישרים עם f (סדר: משמאל עליון לימין תחתון)

שמאל תחתון: נקודות החיתוך של הפונקציה f עם הישרים הרבים שמצאנו

ימין תחתון: הפונקציה החדשה שנוצרה כהעברת קווים ישרים בין נקודות החיתוך

מסקנה

בעבודה זו הצגנו את Dynamic, פונקציית ההפעלה המותאמת לצרכי רשת ונתונים שונים לצורך למידה היעילה יותר ביחס לפונקציית ReLU בעקבות היותה פונקציה כללית. לדעתנו קיים פוטנציאל אמיתי בהמשך מחקר על פונקציות אשר מתאימות עצמן לנתונים של הרשת, המודגם על ידי שיפור לא זניח של תוצאות הלמידה על גבי רשת הנוירונים. אף על פי זאת, נראה כי דרוש למצוא דרך לשפר את זמן הריצה של מציאת הפרמטרים הטובים ביותר, מכיוון שרשתות לרוב יעבדו עם הרבה יותר נתונים ומחלקות אשר עליהן צריך ללמוד, ויכילו יותר שכבות מאשר המודל והנתונים שעליהן ביצענו את המחקר.

קישור לגיט שבו נמצא כל קוד המקור -

https://github.com/EvjaG/DeepLearningProject_2022_Azrieli

ניתן למצוא חומר נוסף וקישורים משלימים בנספח המצורף להגשת העבודה