פרויקט סיום בקורס Deep Learning

שאלה 1 – בניית מודל משופר וניסויים על היפר-פרמטרים שונים.

לקחנו את הרשת מתרגיל 2 ושיפרנו אותה לפי השיפורים שהצענו בסוף התרגיל.

עשינו 3 ניסויים על הפרמטרים הבאים:

היפר-פרמטר	ערכים שנבדקו	הערך שנבחר
Layers_per_block	8,11,16,22	8
Hidden_dims	**	**
Hidden_layers_size	2,5,10	5
Hidden_layers_count	70,100,120	100

*את הפרמטר filters_per_layers בחרנו לפי תוצאות ניסוי שביצענו בתרגיל הקודם, עם הערך 64.

**את הפרמטר hidden_dims פיצלנו לשני פרמטרים: Hidden_layers_size, Hidden_layers_count

***בניסוי של Hidden_layers_count החלטנו לבדוק את הערכים על שני Hidden_layers_size- 2 ו-5

שאלה 2 - ניסויי learning rate ואופטימיזציות

על כל אחת מהאופטימיזציות הבאות: sgd, SGDMomentum, AdaGrad, ADAM

ננסה למצוא ערך learning_rate מיטבי. עבור כל אחד נבדוק את הערכים:

0.1, 0.01, 0.005, 0.001, 0.0005

עבור SGDMomentum לא היתה למידה על הvalidation והtest כלל (כן היה מעט על הtrain).

עבור SGD ללא אופטימיזר היתה למידה רק עם learning rate של 0.1, ועם 0.01 היתה למידה מאוד איטית שלא הגיעה למיצוי הלמידה בניסוי.

AdaGrad היה מיטבי עם ערך learning rate 0.005 והגיעו בערך ל65%

ADAM היה מיטבי עם ערך learning rate 0.001 והגיע בערך ל65%

בהשוואה בין שני תוצאות הtest הגענו לכך שAdaGrad מגיע (בהבדל קטן) לתוצאות טובות יותר מ ADAM.

שאלות מהמחברת:

?Why should we add the momentum to SGD

מומנטום מוסיף לחישוב הצעד את היסטוריית הצעדים הקודמים כך שצעדים מוקדמים משפיעים פחות על הצעד הנוכחי ואילו הצעדים האחרונים משפיעים יותר. השפעה זו גורמת לכך שהצעד הנוכחי מקבל כיוון וגודל אחרים מאשר ללא המומנטום.

בעזרת המומנטום ניתן לבצע צעדים בכיוון "נכון" יותר ובכך "לדלג" על נקודות מינימום מקומיות ובאופן כללי להגיע למינימום מהר יותר.

?What is the main two improvments adam has comparing to the simple SGD

אדם מכיל בתוכו את האופטימייזרים momentum, RMSprop כאשר מומנטום משלב היסטוריה ומכוונן בהתאם וRMSprop מנרמל את הכיוון בהתאם לסדר גודל הגרדיאנטים.

?What was the best optimization algorithm and best learning rate you've found

מהניסויים שערכנו הגענו לכך שAdaGrad optimizer עם ערך 0.005 learningRate היה המוצלח ביותר.

הערה:

לאחר ההרצות הנ"ל למציאת הlearning rate המיטבי, החלטנו להוסיף לרשת skip connections ולכן הרצנו את הניסויים מחדש על מנת לראות האם חל שינוי על הפרמטרים אותם בדקנו.

בנוסף הבנו כי אין צורך להקטין את הvalidatoin set כי אנחנו עלולים לבחור פרמטרים שמכלילים פחות טוב וההרצה של הבדיקה לא ארוכה משמעותית.

הגענו לכך שעלינו לשנות את הפרמטרים שבחרנו לסט פרמטרים הבא:

היפרפרמטר	הערך לפני השינוי	הערך שנבחר אחרי
Layers_per_block	8	4
Hidden_dims	**	**
Hidden_layers_size	5	4
Hidden_layers_count	100	120

* Hidden_layers_size- בדקנו על הפרמטרים 2, 4, 6

*כמו כן החלטנו לשנות את הפרמטר pool_every (מספר השכבות שלאחריו עושים pooling) מ8 ל2

שאלה 3 – batch normalization

בסוף התרגיל הקודם הצענו הצעה לשיפור המודל, להוסיף batch normalization ובתחילת הפרויקט כשיצרנו את המודל מהמטלה הקודמת, כבר הוספנו את תוספת זו ובעצם ביצענו את כל השאלות עד כה כבר עם batch normalization.

שאלות מהמחברת:

What is the purpose of batch normalization (why do we use it)?

השימוש בbatch normalization פותר בעיה שrelu יוצרת והיא איפוס משקולות לכן בעזרת הנרמול ניתן לאמן רשתות עמוקות יותר.

בנוסף הנרמול שומר על הגרדיאנטים באותו נורמל ולכן האימון מהיר יותר.

Did the Batch normalization improve the network preformance?

הראינו בתרגיל קודם כי התוספת של batch normalization משפר את הרשת ומאיץ למידה.

שאלה 4 – regularization

על מנת להשוות בין הרגולציות l1,l2 הרצנו את המודל

1. ללא רגולריזציה כלל.
2. עם רגולריזציה L1 וערכי alpha שונים.
3. עם רגולריזציה L2 וערכי alpha שונים.

נבדוק את ערכי אלפא: 0.1, 0.01, 0.001, 0.005, 0.0005

תוצאות:

מודל בלי רגולריזציה מצליח ללמוד אך מגיע לoverfit יחסית מהר ולא מגיע לביצועים טובים.

עבור L1 נגיע לתוצאות טובות ביותר עם 0.01

עבור L2 נגיע לתוצאות טובות ביותר עם 0.005

וסהכ הרגולריזציה L2 היא המוצלחת ביותר.

שאלות מהמחברת:

Why should we use regularization?

הרגולריזציה מגבילה את הנורמל של המשקולות ובכך מונעת מצב בו הם מגיעות לערכים גדולים עד לחריגות. כמו כן היא גם עוזרת למנוע overfitting על ידי הקטנת טווח הפונקציות שהמודל בודק ובכך עוזרת גם להכללה.

How does the regularization affect the train accuracy and loss?

ראינו בהרצות כי הוספת רגולריזציה משפיעה על קצב עליית הtrain accuracy- הוא יותר איטי ויציב

וכן גם ראינו כי הtrain loss יורד עם פחות קפיצות חדות.

?How does it affect the val and test accuracy and loss

הval accuracy וval loss מושפע באופן דומה להשפעות שיש על train

ניתן לראות כי עם רגולריזציה הביצועים על הtest יותר טובים וכן ללא רגולריזציה כלל המודל לא מצליח ללמוד.

?What was the best regulrization method

לפי הניסויים שערכנו L2 היתה רגולריזציה מוצלחת יותר עם ערך alpha 0.01

<u>שאלה 5 –</u> אוגמנטציות

כדי לשפר את המודל על ידי הוספת אוגמנטציות, נוסיף עבור כל דוגמה דוגמה זהה עליה נבצע אוגמנטציות באופן הבא:

נבחר רנדומלית בהסתברות של 0.3 מבין האוגמנטציות הבאות-

colourJitter- משפיע ומשנה את צבעי התמונה באופן רנדומלי.

randomRotation- מסובב את התמונה לזווית רנדומלית בין -45 ל45 מעלות.

HorizontalFlip- הופך את התמונה לתמונת המראה שלה.

שאלה מהמחברת:

Did the data augmentation improve the model preformace?

הביצועים של המודל ירדו לאחר הוספת האוגמנטציות.

Mention which augmentation you used and why.

בחרנו באוגמנטציות אלו כי רצינו לוודא שהאובייקט שצריך לזהות בתמונה ישאר לאחר השינוי ועדיין יהיה ברור לזיהוי. שינוי צבע, סיבוב, ומראה לא אמורים לאבד את האובייקט המרכזי בתמונה.

סיכום חלקי –

What is your best architecture?

אלו הפרמטרים שנבחרו לאחר כל הניסויים:

filter_count=64

pool_every = 2

layers_per_block = 4

hidden_layer_size = 120

hidden_layer_count = 4

batch_size = 256

epoch_num=100

lr=5e-3

reg=5e-3

optimizer="AdaGrad"

l1 = False

מבחינת מבנה הרשת:

הרשת בנויה מבלוקים שביניהם יש שכבת דרופאאוט. אחרי ארבעה בלוקים עוברים לרשת FC המכילה 4 שכבות חבויות של 120.

בכל בלוק יש שני מסלולים-

במסלול הראשון יש קונבולוציה שמעבירה את הקלט מכמות הצ'אנלים בקלט לכמות הצ'אנלים בפלט ועם קרנל של 1X1, ובמסלול השני עושים אקטיבציה-נרמול-קונבולציה(עם קרנל 3X3)-אקטיבציה-קונבולציה (פונקציית אקטיבציה leakyRelu). ובסוף מחברים בין שני המסלולים.

כל בלוק חוזר 4 פעמים כאשר לאחר כל שניים מתבצע pooling שמקטין את התמונה פי 2.

what its best accuracy and loss on the test set?

Best accuracy: 80.03%

Best loss: 0.609

Compare your result to assignment 2. How did you mange to improve the model?

הצלחנו לשפר את המודל בכך שדייקנו את ההיפר-פרמטרים ולקחנו את כל הערכים המיטביים מכל פרמטר וכן שינינו את הארכיטקטורה.

סיכום בלוג –

:Read about ResNet in this blog

https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035

:**Summarize this blog in the report.** Be sure you answer this question

1. What is the 'res' stand for in resnet?

   Residual Network

2. What is the main innovative idea presented in resnet?

החידוש המרכזי שהגיע עם resnet הוא הרעיון של מעקף וחיבור בין שכבות חבויות באופן **רציף**.

3. which probelm this unique architecture tring ro solve?

קיימת בעיה לאמן רשתות עמוקות בגלל שגרדיאנטים הולכים ונמתקרבים ל0 עקב הכפלות הולכות ונשנות בתהליך חלחול הגרדיאנטים, ועל בעיה זו resnet באה להתגבר.

סיכום הבלוג:

Resnet היא משפחה של רשתות נוירונים עמוקות שמתרכזות בלימוד ההבדלים בין שכבות.
מארכיטקטורות שונות שהיו קיימות עד שפורסמה resnet, ניתן היה להבין כי רשתות עמוקות נוטות להגיע לביצועים טובים יותר ונמנעות מoverfit, אך עם זאת האימון שלהן איטי וכן נתקלים בבעיית הגרדיאנט הנעלם.

Resnet באה לפתור בעיה זו על ידי הוספת קישורים בין שכבויות חבויות על ידי הוספת שכבות זהות.
המפרסמים של resnet טענו שהוספת חיבורים אלו שקולה להוספת שכבות זהות לרשת רדודה יותר ולכן לא אמורה לפגוע בביצועי הרשת על סט האימון.

ישנם מספר רעיונות לשיפור resnet-

ResNeXt: רשת שבה מפצלים את הבלוקים של resnet למספר מסלולים הפועלים באותו אופן. מספר המסלולים הוא היפרפרמטר נוסף הנקרא cardinality.

DenseNet: רשת בה קלט של כל שכבה מורכב מפלט של כל השכבות מלפניה על ידי שרשור. שינוי זה מאפשר אימון מהיר יותר ומעודד שימוש חוזר בפיצ'רים קודמים שכבר נלמדו. קבוצת שכבות משורשרת נקראת dense block.

רשתות מסוג זה מגיעות לביצועים טובים אך דורשות אימון ממושך ולכן לא פרקטיות לשימוש על משימות בזמן אמת. פתרון מוצע הוא באופן רנדומלי לכבות בלוקים בזמן אימון. בלוק כבוי יעביר רק את הקלט שלו לבלוק הבא ללא שינוי דרך שכבת הזהות. פתרון זה מקביל לאימון של כמה רשתות קטנות. אימון בשיטה זו מוכח כמהיר יותר ומגיע לביצועים טובים יותר. עובדה זו מרמזת על כך שישנם בלוקים שיכולים להיות מיותרים.

ההסתברות של בלוק להיות כבוי תואמת למיקום שלו ברשת כך שבלוקים בשכבות נמוכות יותר יכבו בסבירות נמוכה יותר שכן הם מזהים פיצ'רים בסיסיים יותר.

הרעיון מאחורי שיטת אימון זו הוא שכל בלוק מספק שני נתיבים אפשריים למידע כך שעבור n בלוקים יש 2^n נתיבים ברשת, וכיבוי בלוק עדיין משאיר הרבה נתיבים פעילים.

מניסויים ניתן לראות שרוב הנתיבים תורמים לפתרון מכך שרואים שמספר השכבות הכבויות תואם ל error rate.

מחקר נוסף מראה שרוב הגרדיאנטים מגיעים מנתיבים "קצרים" יותר (עוברים במסלולים דרך בלוקים שלא משפיעים על הקלט), ולכן resnet לא באמת פותרת את בעיית הגרדיאנטים הנעלמים אלא רק מקצרת את המסלול בו הגרדיאנטים עוברים ברשת.

שאלה 6 –

שאלה מהמחברת:

Pretrained = false

What was ResNet50 accuracy and loss?

Accuracy- 70.28%

Loss- 1.178

Is it overfit/underfit/well-fit the data?

overfit

Pretrained = true

What was the pretrained ResNet50 accuracy and loss?

Accuracy- 79.28%

Loss- 0.675

Is it overfit/underfit/well-fit the data?

overfit

Has it got better accuracy than the non-pretrained ResNet?

Yes.

Extra Points

בדקנו את המודל denseNet (עליו קראנו בבלוג שסיכמנו) עם משקולות מאומנות מראש, שינינו את השכבת FC האחרונה כך שתתאים לבעיה ותייצר שכבת פלט בגודל 10. הגענו לתוצאות:

Accuracy- 80.64%

Loss- 0.6086

What was the best pretrained architecture?

denseNet

Is it overfit/underfit/well fit the data?

Well fit.