מדריך להתחלת פרויקט חדש

בשמחה אסכם עבורך את תתי-הפרקים מהמדריך להקמת פרויקט למידת מכונה חדש, כפי שהם מופיעים בעמוד:

1. הנחות יסוד (דרישות מוקדמות)

לפני שמתחילים בבחירת הארכיטקטורה והכוונון, עליך לוודא ש:

- כבר התנסית בבעיה והכנת את נתוני האימון.
- הגדרת צינור עיבוד נתונים (Pipeline) לאימון ולבדיקה.
- בחרת והטמעת מדדים (Metrics) שמייצגים נאמנה את מה שתרצה למדוד בסביבת הייצור.

2. בחירת הארכיטקטורה של המודל

- **התחלה בטוחה:** מומלץ להשתמש בארכיטקטורה קיימת שעובדת או במודל מקודד-מפענח (Encoder-Decoder) מסורתי.
- **אל תמציא את הגלגל:** כדאי להתחיל עם מודל ותיק ונפוץ ורק בשלב מאוחר יותר לבנות מודל מותאם אישית.
- **נקודת מוצא:** חפש בסיס קוד (Codebase) מתועד שפתר בעיה דומה ככל האפשר לבעיה שלך ושחזר אותו.

3. בחירת כלי האופטימיזציה (Optimizer)

- **פופולריות היא מפתח:** התחל עם הכלים הנפוצים ביותר לסוג הבעיה שלך, כמו **Adam** או **SGD עם מומנטום** (רצוי וריאנט Nesterov).
- **פשטות בשלבים הראשונים:** בתחילת הפרויקט, בחר אופטימיזציה עם מינימום היפר-פרמטרים לכוונון (למשל, Adam עם ערכי Beta ו-Epsilon קבועים).
- **הדרגתיות:** רק בשלבים מאוחרים יותר כדאי לעבור לכלי אופטימיזציה כלליים יותר ולכוונן את כל הפרמטרים שלהם.

4. בחירת גודל הקבוצה (Batch Size)

- **יעילות ולא ביצועים:** גודל האצווה קובע את **מהירות האימון** וצריכת המשאבים, ולא נועד לכוונן ישירות את דיוק המודל.
- **הכלל המנחה:** לרוב, כדאי לבחור את גודל הקבוצה **הגדול ביותר** שהחומרה שלך (הזיכרון של המאיץ) יכולה להכיל.
- **מניעת צווארי בקבוק:** אם הגדלת גודל הקבוצה לא משפרת את תפוקת האימון (מספר דוגמאות בשנייה), ייתכן שיש בעיה בצינור הקלט/פלט (I/O) שיש לפתור.
- **יתרון:** גודל קבוצה גדול מקצר את זמן האימון, מה שמאפשר לבדוק יותר רעיונות בזמן נתון.

5. קיצור זמן האימון

- זמן האימון מוגדר כמכפלה של הזמן לכל שלב במספר הצעדים הכולל.
- הגדלת גודל הקבוצה בדרך כלל מפחיתה את מספר השלבים הכולל הנדרש להגעה ליעד, ובכך מקצרת את זמן הפיתוח הכולל.

גישה מדעית לשיפור ביצועי המודל

להלן סיכום מפורט של תתי-הפרקים מהעמוד העוסק בגישה מדעית לכוונון מודלים של למידה עמוקה, תוך דגש על הנקודות המרכזיות בכל חלק:

1. אסטרטגיית הכוונון המצטבר (Incremental Tuning Strategy)

- **התחלה פשוטה:** מומלץ להתחיל עם הגדרה בסיסית ופשוטה ולהוסיף שיפורים בהדרגה.
- **צבירת תובנות:** המטרה היא לא רק לשפר את התוצאה, אלא להבין את הבעיה לעומק תוך כדי התהליך.
- **ביסוס על ראיות:** אין לאמץ שינוי או להוסיף מורכבות לצינור האימון אלא אם יש הוכחות חזקות לכך שהשיפור אינו מקרי.
- **מגבלות האוטומציה:** חיפוש אוטומטי מלא אינו מעשי בגלל גודל מרחב האפשרויות; לכן, יש להשתמש באלגוריתמים אוטומטיים בתוך מרחבי חיפוש שמוגדרים ומעודכנים על ידי בני אדם.

2. ארבעת השלבים של סבב הניסויים

האסטרטגיה מבוססת על חזרה על מחזור קבוע:

1. **בחירת יעד:** הגדרת מטרה ממוקדת לסיבוב הנוכחי.
2. **תכנון:** בניית סדרת ניסויים להשגת היעד.
3. **למידה:** ניתוח התוצאות והערכתן מול רשימת משימות.
4. **החלטה:** קביעה האם לאמץ את השינוי המוצע כסטנדרט החדש ("השקה").

3. בחירת יעד לסיבוב הניסויים הבא

- **מיקוד:** יש לבחור יעד בהיקף מתאים כדי שניתן יהיה לבודד את ההשפעה של כל שינוי (למשל: בדיקת רגולריזטור חדש או הבנת השפעת היפר-פרמטר ספציפי).
- **טווח ארוך מול קצר:** עדיף להשיג הבנה שתשרת את הפרויקט לטווח ארוך מאשר להשיג שיפור קטן וזמני בשגיאת האימות.

4. התמקדות בתובנות במקום ברווחים לטווח קצר

- **חיפוש (Exploration) מול ניצול (Exploitation):** כדאי להקדיש את רוב הזמן להבנת מבנה בעיית ההתאמה ורק חלק קטן למקסום הביצועים הסופי.
- **יתרונות ההבנה:** הבנה עמוקה עוזרת לזהות אילו פרמטרים הם הרגישים ביותר, אילו תכונות מיותרות וניתן להסירן, ומתי כוונון נוסף הגיע לנקודת מיצוי.
- **צמצום מרחב החיפוש:** ככל שההבנה גדלה, ניתן לצמצם את מרחב החיפוש סביב הערכים האופטימליים ולשפר את יעילות העבודה.

5. תכנון סבב הניסויים הבא

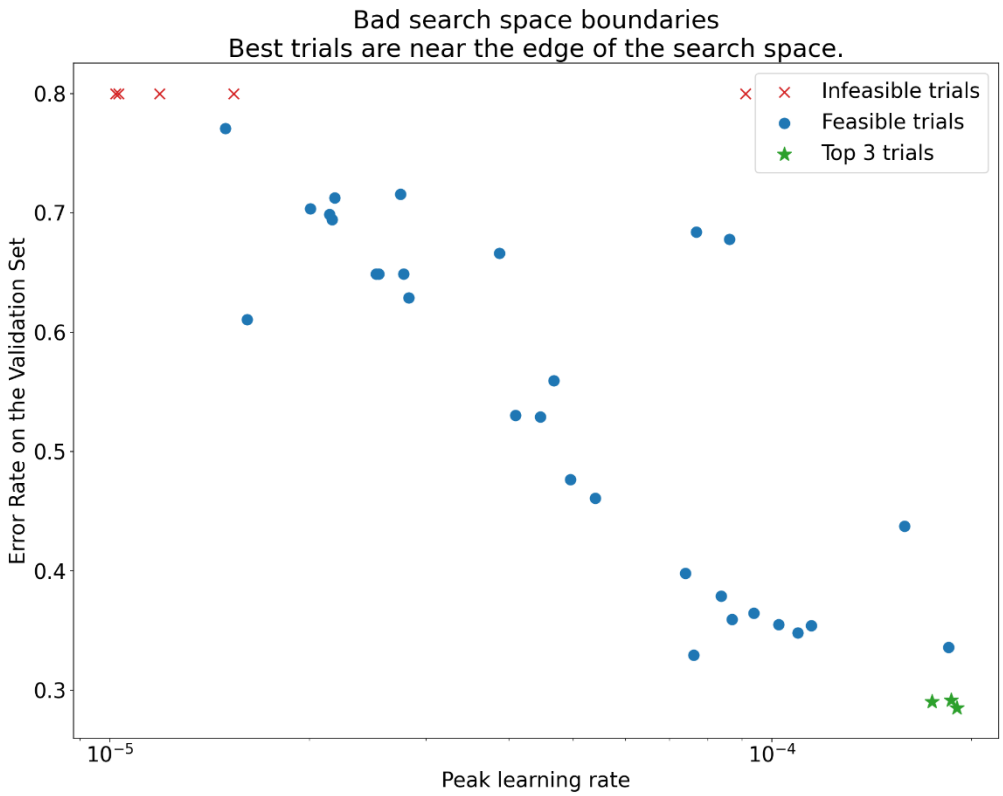
- **סיווג פרמטרים:** יש להגדיר מראש אילו היפר-פרמטרים הם "מדעיים", אילו "מפריעים" ואילו "קבועים" (פירוט בתת-הפרק הבא).
- **אופטימיזציה הוגנת:** כדי להשוות בין ערכים שונים של פרמטר מדעי, יש לבצע אופטימיזציה של הפרמטרים המפריעים עבור כל אחד מהם.
- **איזון משאבים:** בחירת מרחב החיפוש צריכה לאזן בין עלות המחשוב לבין הערך המדעי שמתקבל.

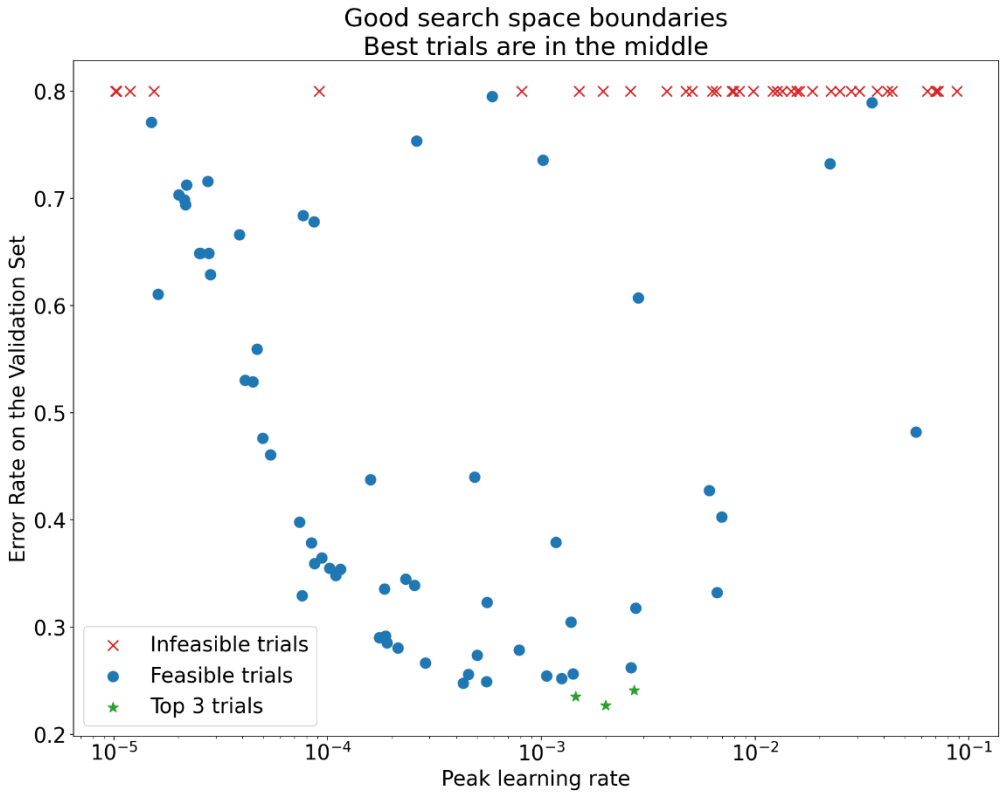
6. זיהוי היפר-פרמטרים מדעיים, מפריעים וקבועים

הסיווג של כל פרמטר תלוי ביעד הניסוי:

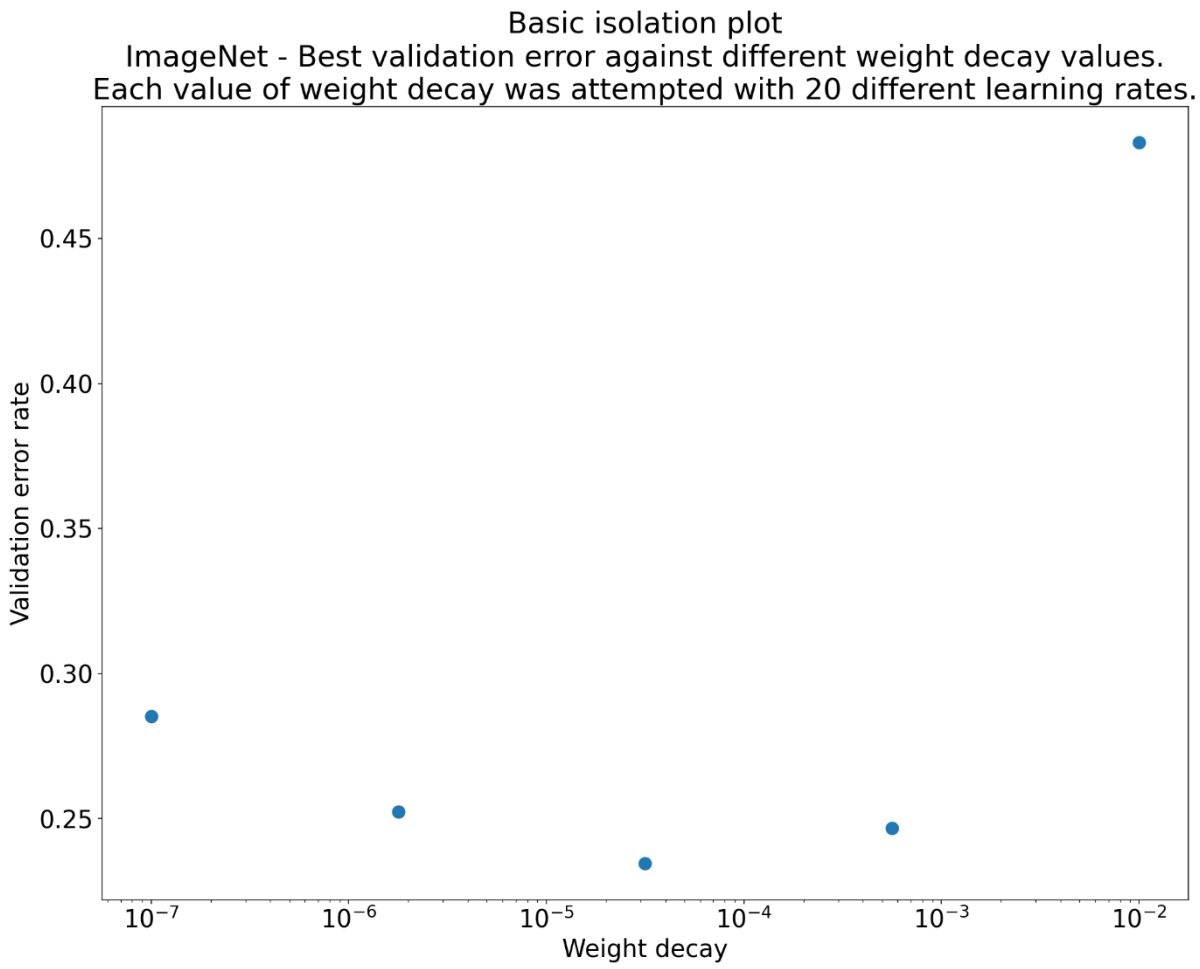
- **היפר-פרמטרים מדעיים:** אלו שאת השפעתם הישירה אנו רוצים למדוד (למשל: "האם ReLU טוב יותר מ-tanh?").
- **היפר-פרמטרים מפריעים (Nuisance):** פרמטרים שחייבים להשתנות כדי לאפשר השוואה הוגנת. לדוגמה, כשמשנים את מספר השכבות (מדעי), חייבים לכוונן מחדש את קצב הלמידה (מפריע) כי הערך האופטימלי שלו משתנה.
- **היפר-פרמטרים קבועים:** ערכים שנקבעו מראש ולא משתנים בניסוי הנוכחי. חשוב לזכור שהמסקנות מהניסוי תקפות רק תחת המגבלות של הערכים הקבועים הללו.

אני מקווה שסיכום זה מספק לך תמונה ברורה של התוכן. אם תרצה להעמיק באחד הנושאים, אני כאן.





איור 1: דוגמאות לגבולות לא תקינים של מרחב חיפוש ולגבולות תקינים של מרחב חיפוש.



איור 2 :תרשים בידוד שבודק את הערך הטוב ביותר של דעיכת המשקל עבור ResNet-50 שאומן על ImageNet.

קובעים את מספר השלבים לכל הרצת אימון

להלן סיכום של תתי-הפרקים מהמדריך לכוונון למידה עמוקה בנושא מספר צעדי האימון (Number of Steps), עם הנקודות המרכזיות מכל אחד:

1. שני סוגים של עומסי עבודה של אימון

הפרק מבחין בין שני מצבים עיקריים של אימון מודלים:

- **אימון מוגבל-חישוב (Compute-bound):** המגבלה היא זמן האימון שניתן להקדיש (תקציב). במצב זה, זמן האימון האופטימלי הוא פשוט "כמה שיותר". שיפור היעילות או הארכת הזמן ישפרו ישירות את התוצאות.
- **אימון שאינו מוגבל בחישוב:** ניתן לאמן כמה שרוצים, אך אימון ארוך מדי עלול להוביל להתאמת יתר (Overfitting). במצב זה קל יותר לבצע אופטימיזציה של לוחות זמנים של שיעור הלמידה.
- **גורמים המאטים את האימון:** שימוש בגודל קבוצה (Batch size) קטן, הוספת הגדלת נתונים (Data augmentation) ורגולריזציה (כמו Dropout) מגדילים את שונות הגרדיאנטים ומצריכים יותר צעדי אימון.

2. החלטה לגבי משך האימון כשאינו מוגבל במשאבים

המטרה היא להגיע לתוצאה הטובה ביותר בלי לבזבז משאבים מיותרים.

- **כלל אצבע:** אם יש ספק, עדיף להתאמן יותר זמן.
- **ניהול ניסויים:** אין לשנות את max_train_steps במהלך מחקר (Study). יש לקבוע ערך קבוע לכל הניסיונות.
- **ניתוח תוצאות:** אם נקודת הביקורת (Checkpoint) הטובה ביותר נמצאת תמיד ב-10% הראשונים של האימון, הערך המקסימלי גבוה מדי. אם היא נמצאת ב-25% האחרונים, כדאי להאריך את האימון.
- **שינויים דינמיים:** הוספת רגולריזציה או הגדלת נתונים דורשת לרוב הגדלה של מספר הצעדים.

3. אלגוריתם לבחירת מועמד ראשוני ל-max_train_steps

שיטה למציאת נקודת התחלה למספר הצעדים באמצעות סריקת שיעור למידה (Learning Rate Sweep):

- **הנחה:** ניתן להגיע להתאמה מושלמת (Perfect fit) לנתוני האימון עם קצב למידה קבוע.
- **התהליך:** מריצים חיפוש על קצב הלמידה ללא רגולריזציה או הגדלת נתונים. מספר השלבים שנדרש כדי להגיע לביצועים מושלמים בחיפוש זה יהיה הניחוש הראשוני (N).
- **אזהרה:** יש לוודא שמרחב החיפוש של קצב הלמידה אינו קטן מדי, כדי לא להסיק בטעות שצריך מספר צעדים גבוה מאוד.

4. החלטה לגבי משך האימון כשהוא מוגבל במשאבים

במצב זה, האימון יכול להשתפר כמעט ללא הגבלה, ולכן נדרשת אסטרטגיה לניהול זמן:

- **איזון:** עדיף להריץ הרבה ניסויים קצרים לכוונון (Tuning) מאשר מעט ניסויים ארוכים מאוד.
- **גישה הדרגתית:** מומלץ לבצע אופטימיזציה בכמה סבבים (1-3 סבבים). מתחילים בניסויים קצרים מאוד כדי לקבל תובנות מהירות, ובהדרגה מגדילים את זמן האימון עבור המודלים המבטיחים ביותר.
- **רלוונטיות:** תובנות מניסויים קצרים (למשל 10% מזמן הייצור) לא תמיד תקפות לריצות ארוכות, ולכן יש לבדוק מחדש את המסקנות ככל שמאריכים את זמן האימון.

הנחיות נוספות לצינור עיבוד הנתונים לאימונים

אסכם עבורך את תתי-הפרקים בעמוד העוסק בהנחיות נוספות לכיוונון (tuning) של למידה עמוקה. להלן הנקודות החשובות מכל פרק:

1. אופטימיזציה של צינורות עיבוד נתונים (Pipeline Optimization)

תת-פרק זה מתמקד בזיהוי ופתרון צווארי בקבוק בשלב הזנת הנתונים למודל.

- **אבחון:** יש להשתמש בכלי פרופיל (כמו Perfetto ל-JAX או TensorFlow profiler) כדי לזהות אם צינור הנתונים מעכב את האימון.
- **סיבות נפוצות לעיכובים:** נתונים שאינם מאוחסנים קרוב לתהליך האימון (זמן אחזור ברשת), עיבוד מקדים (preprocessing) יקר שמתבצע "אונליין", ומחסומי סנכרון מיותרים.
- **פתרונות מוצעים:**
  - שימוש ב-**Prefetching** (אחזור מראש) כדי להכין דוגמאות בזמן שהמודל מתאמן.

- הסרת תכונות (features) ומטא-נתונים מיותרים בשלב מוקדם.
- הגדלת מספר המשימות שיוצרות דוגמאות (למשל באמצעות שירות tf.data).

2. הערכת הביצועים של המודל (Evaluating Model Performance)

תת-פרק זה מסביר כיצד למדוד את איכות המודל בצורה נכונה.

- **סוגי הערכה:** קיימת הערכה "אונליין" (בייצור), "אופליין" (על מערכי נתונים קבועים) והערכות תקופתיות במהלך האימון.
- **הגדרת הערכות תקופתיות:**
  - מומלץ לבצע הערכות לפי **מרווחי שלבים קבועים** (steps) ולא לפי זמן, כדי למנוע הטיות עקב תקלות רשת או עיכובים טכניים.
  - גודל האצווה (batch size) בהערכה יכול להיות גדול יותר מאשר באימון.
  - חשוב לוודא שדוגמאות "ריפוד" (padding) לא מטהות את התוצאות (לתת להן משקל אפס).
- **דגימה להערכה:** אם מערך הנתונים גדול מדי, יש לדגום אותו בזהירות כדי לוודא שהמדגם מייצג ולא מוטה, במיוחד במקרים של מערכי נתונים לא מאוזנים.

3. שמירת נקודות ביקורת (Checkpointing) ובחירה רטרוספקטיבית

תת-פרק זה עוסק בבחירת הגרסה הטובה ביותר של המודל.

- **אסטרטגיה:** מריצים את האימון למספר שלבים קבוע מראש ושומרים נקודות ביקורת (checkpoints) לאורך הדרך.
- **בחירה רטרוספקטיבית:** הנקודה הטובה ביותר היא לא תמיד האחרונה. מומלץ לעקוב אחרי N נקודות הביקורת הטובות ביותר שנראו במהלך כל הריצה, ובסיום לבחור את הטובה מכולן.
- גישה זו מייתרת לרוב את הצורך ב"עצירה מוקדמת" (early stopping) מורכבת.

4. הגדרת מעקב אחר ניסויים (Experiment Tracking)

תת-פרק זה מדגיש את החשיבות של סדר וארגון בניהול הניסויים.

- **מה לתעד:** חשוב לעקוב אחרי הביצועים הטובים ביותר של נקודות הביקורת בכל מחקר ולשמור תיאור קצר של מטרת הניסוי.
- **כלי עבודה:** ההמלצה הפשוטה והיעילה היא להשתמש בגיליון אלקטרוני (spreadsheet) כדי לרכז את כל תוצאות הניסויים השונים.

קצב למידה

אסכם עבורך את תתי-הפרקים המופיעים בעמוד העוסק בקצב למידה (Learning Rate) מתוך המדריך לכוונון למידה עמוקה:

1. תזמון של ירידה בקצב הלמידה (Learning Rate Decay Scheduling)

- **הבעיה:** מציאת הלו"ז (Schedule) האופטימלי להפחתת קצב הלמידה היא עדיין בעיה פתוחה במחקר.
- **החשיבות:** למרות שאין "נוסחת קסם", חשוב מאוד להשתמש בלוח זמנים משתנה ולא בקצב קבוע.

- **הסיבה:** שלבים שונים בתהליך האופטימיזציה דורשים קצבי למידה שונים כדי להגיע לתוצאות טובות.

2. הדעיכה הטובה ביותר המוגדרת כברירת מחדל

- **המלצה:** כברירת מחדל, מומלץ להשתמש באחת משתי המשפחות הבאות:
  - **דעיכה לינארית (Linear Decay).**
  - **דעיכה קוסינוסית (Cosine Decay).**
- קיימות משפחות נוספות שעשויות להתאים, אך אלו הן נקודות הפתיחה המומלצות.

3. למה בחלק מהמאמרים יש תוכניות מורכבות של שיעורי למידה?

- **מקור המורכבות:** תוכניות מסובכות במאמרים הן לרוב תוצאה של כוונון ידני (Ad-hoc) – החוקרים עוצרים את האימון כשהביצועים נעצרים, מפחיתים את הקצב ידנית וממשיכים.
- **המלצה:** לא מומלץ להעתיק לוחות זמנים כאלו, כיוון שהם תלויים מאוד בבחירות אחרות של היפר-פרמטרים במודל הספציפי.
- **שחזור:** קשה מאוד לשחזר תוצאות שהתקבלו מהתערבות אנושית כזו. עדיף להשתמש בתזמון אוטומטי לחלוטין.

4. איך כדאי לכוון את ההיפר-פרמטרים של Adam?

העמוד מציע סדר עדיפויות לכוונון בהתאם ל"תקציב" הניסויים שלכם:

- **פחות מ-10 ניסויים:** כוונו רק את **קצב הלמידה הבסיסי**.
- **10-25 ניסויים:** כוונו את קצב הלמידה ואת **beta_1**.
- **יותר מ-25 ניסויים:** כוונו את קצב הלמידה, **beta_1** ו-**epsilon**.
- **הרבה יותר מ-25 ניסויים:** ניתן להוסיף גם את כוונון **beta_2**.

חיפוש מעשי אקראי

אשמח לסכם עבורך את תתי-הפרקים בעמוד העוסק בחיפוש כמעט אקראי (Quasi-random search) מתוך המדריך לכוונון למידה עמוקה.

להלן סיכום הנקודות החשובות מכל חלק:

1. למה כדאי להשתמש בחיפוש כמעט אקראי?

- **שלב החקירה:** השיטה מועדפת בשלב שבו רוצים למקסם תובנות לגבי בעיית הכוונון (Exploration), בעוד ששיטות כמו אופטימיזציה בייסיאנית מתאימות יותר לשלב הניצול (Exploitation).
- **פיזור אחיד:** חיפוש כמעט אקראי מפזר את נקודות הדגימה בצורה אחידה יותר במרחב החיפוש מאשר חיפוש אקראי רגיל, מה שמאפשר כיסוי טוב יותר של האפשרויות.

2. יתרונות על פני כלים מתוחכמים (כמו אופטימיזציה בייסיאנית)

- **ניתוח "אחרי מעשה" (Post-hoc):** מכיוון שהדגימה אינה אדפטיבית (לא תלויה בתוצאות קודמות), ניתן לשנות את יעד האופטימיזציה (למשל, לעבור משגיאת אימות לשגיאת אימון) מבלי להריץ את הניסויים מחדש.

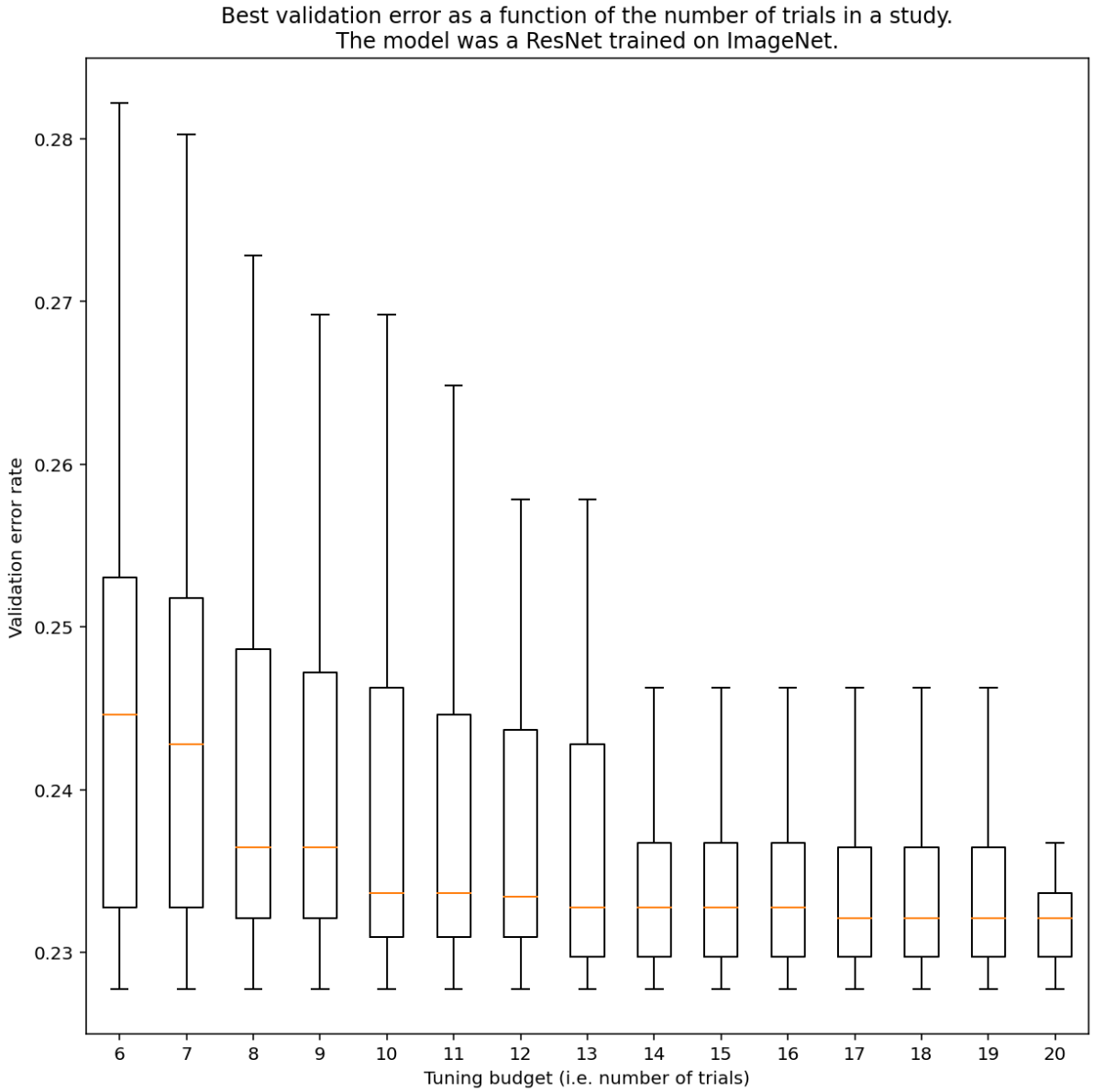
- **עקביות ושחזור:** קל יותר לשחזר מחקרים ישנים גם אם אלגוריתם החיפוש השתנה מעט, כל עוד נשמרים מאפייני האחידות.
- **הבנת מרחב החיפוש:** קל יותר להסיק מסקנות. לדוגמה, אם התוצאה הטובה ביותר נמצאת בקצה טווח החיפוש, זהו רמז ברור שצריך להזיז את גבולות החיפוש.
- **מקביליות:** אין הבדל סטטיסטי בין הרצת ניסויים במקביל לבין הרצתם ברצף.
- **פשטות ועמידות:** השיטה פשוטה ליישום, מטפלת היטב בנקודות לא ריאליות, וקשה מאוד לאלגוריתמים מתוחכמים לנצח אותה כשיש תקציב כפול ומקביליות גבוהה.

3. איפה אפשר למצוא הטמעה של חיפוש כמעט אקראי?

- **Vizier:** ניתן להשתמש בגרסת הקוד הפתוח של Vizier על ידי הגדרת האלגוריתם כ-QUASI_RANDOM_SEARCH.
- **רצף Halton:** ההטמעות המומלצות מתבססות על "רצף Halton" שעבר ערבוב (Scrambled Halton sequence).
- **חלופות:** אם אין גישה לחיפוש כמעט אקראי, חיפוש אחיד פסאודו-אקראי הוא חלופה סבירה, אם כי מעט פחות יעילה.

4. כמה ניסיונות נדרשים כדי לקבל תוצאות טובות?

- **אין מספר קבוע:** הכמות תלויה בבעיה הספציפית, אך העמוד מציג דוגמה של מודל ResNet-50.
- **השפעת התקציב:** בניסוי שהוצג, 6 ניסיונות הראו שונות גבוהה מאוד בתוצאות, בעוד ש-20 ניסיונות סיפקו יציבות רבה יותר.
- **כלל אצבע:** חיפוש אקראי (או כמעט אקראי) עם תקציב כפול נחשב ל"קו בסיס" (Baseline) חזק מאוד שקשה לאלגוריתמים אדפטיביים לנצח, במיוחד בתנאים של הרצת ניסויים רבים במקביל.



איור 3 ResNet-50 :שהותאמה ב ImageNet-עם 100 ניסיונות. באמצעות טעינה ראשונית ,(bootstrapping) בוצעה סימולציה של סכומים שונים של תקציב לכוונון. מוצגים תרשימים של תיבות עם הביצועים הטובים ביותר לכל תקציב ניסיוני.

שאלות נפוצות

בשמחה אסכם עבורך את תתי-הפרקים מהמדריך לניפוי באגים בבעיות אופטימיזציה (מתוך ה- Deep Learning Tuning Playbook).

להלן סיכום הנקודות החשובות מכל תת-פרק:

1. זיהוי עומסי עבודה לא יציבים (Identifying unstable workloads)

הפרק מסביר כיצד לזהות אם המודל סובל מחוסר יציבות שמעכב את האימון:

- **מתי זו בעיה?** חוסר יציבות הוא בעייתי רק אם הוא מאלץ אתכם להשתמש בקצב למידה (Learning Rate) קטן מדי, מה שפוגע בביצועים.
- **סוגי חוסר יציבות:** יש להבחין בין חוסר יציבות המתרחש ממש בתחילת האימון לבין כזה שמופיע פתאום באמצעו.
- **שיטת זיהוי:** מבצעים סריקה של קצבי למידה. אם בקצבים שגבוהים רק מעט מהקצב האופטימלי (lr^*) רואים שההפסד (Loss) עולה במקום לרדת, יש בעיית יציבות.
- **טיפ לאבחון:** מומלץ לתעד את נורמת ה-L2 של הגרדיאנטים כדי לזהות ערכים חריגים שגורמים לקפיצות ב-Loss.

2. תיקונים אפשריים לדפוסי חוסר יציבות נפוצים

כאן מוצגות דרכים פרקטיות לפתרון הבעיות שזוהו:

- **חימום (Warmup):** יעיל במיוחד לחוסר יציבות בתחילת האימון.
- **חיתוך גרדיאנטים (Gradient Clipping):** עוזר גם בתחילת האימון וגם באמצעו, ומטפל ב"אתחולים שגויים".
- **החלפת אופטימייזר:** לפעמים Adam יציב יותר מ-Momentum.
- **שיפור הארכיטקטורה:** הוספת חיבורים שיוריים (Residual connections) ונורמליזציה. חשוב להקפיד על הסדר: $x + Norm(f(x))$.
- **הפחתת קצב הלמידה:** זהו המוצא האחרון אם שום דבר אחר לא עזר.

3. חימום קצב הלמידה (Learning rate warmup)

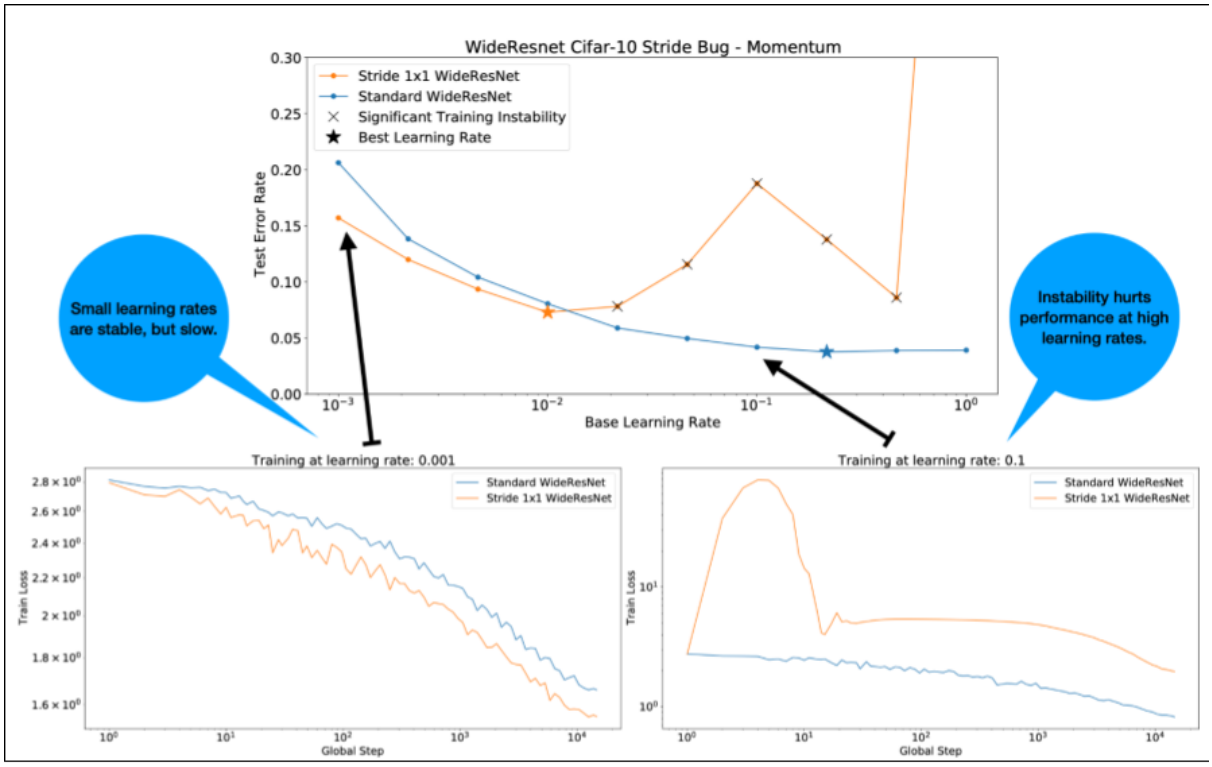
פירוט על טכניקת ה-Warmup:

- **מתי להשתמש?** כאשר קצב הלמידה הטוב ביותר נמצא "על הקצה" של חוסר היציבות (כלומר, קצב מעט גבוה יותר גורם ל-NaN או ל-Loss גבוה מאוד).
- **איך זה עובד?** מעלים את קצב הלמידה בהדרגה מ-0 לערך היעד (Base LR) לאורך מספר צעדים (warmup_steps).
- **המטרה:** למצוא את מספר הצעדים הקצר ביותר שמאפשר להשתמש בקצב למידה גבוה משמעותית מזה שהיה גורם לחוסר יציבות ללא החימום.

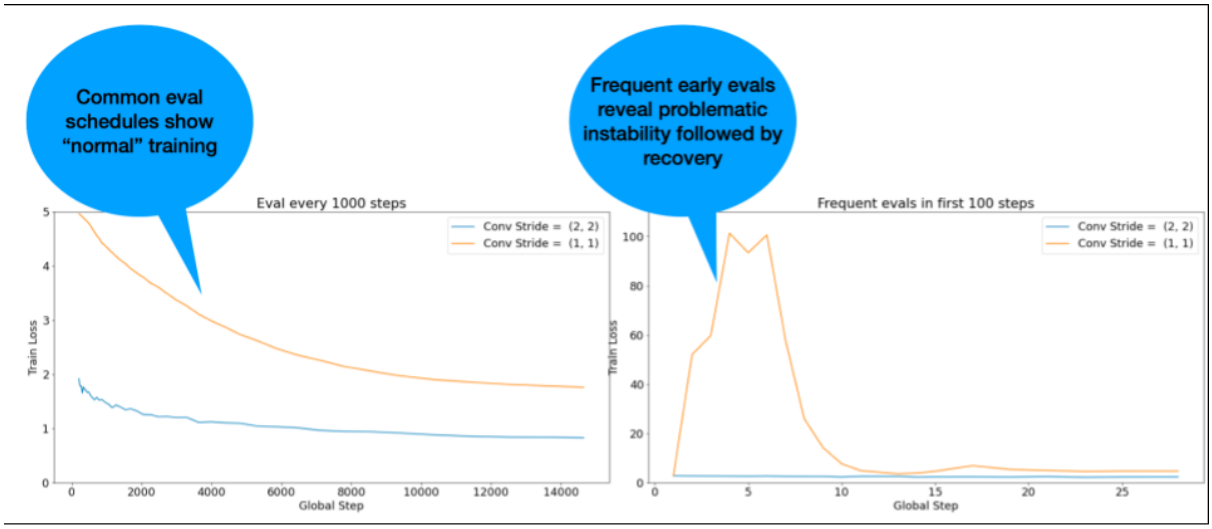
4. חיתוך הדרגתי (Gradient clipping)

הסבר על הגבלת עוצמת העדכונים:

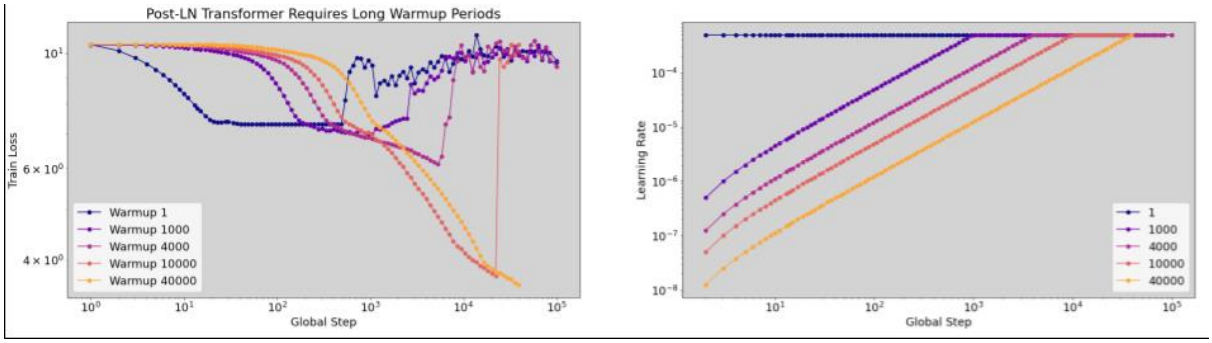
- **התועלת:** מונע מגרדיאנטים חריגים (גדולים מדי) "לפוצץ" את האימון.
- **מתי זה עוזר?** יעיל מאוד נגד קפיצות פתאומיות בגרדיאנט במהלך האימון או נורמות גבוהות מאוד בהתחלה.
- **איך קובעים סף?** הסף האידיאלי לחיתוך (λ) צריך להיות מעט מעל הנורמה ה"טיפוסית" של הגרדיאנטים במודל. אם הנורמה עוברת את הסף, מקטינים את הגרדיאנט באופן יחסי כדי לשמור על הכיוון שלו אך להגביל את גודלו.



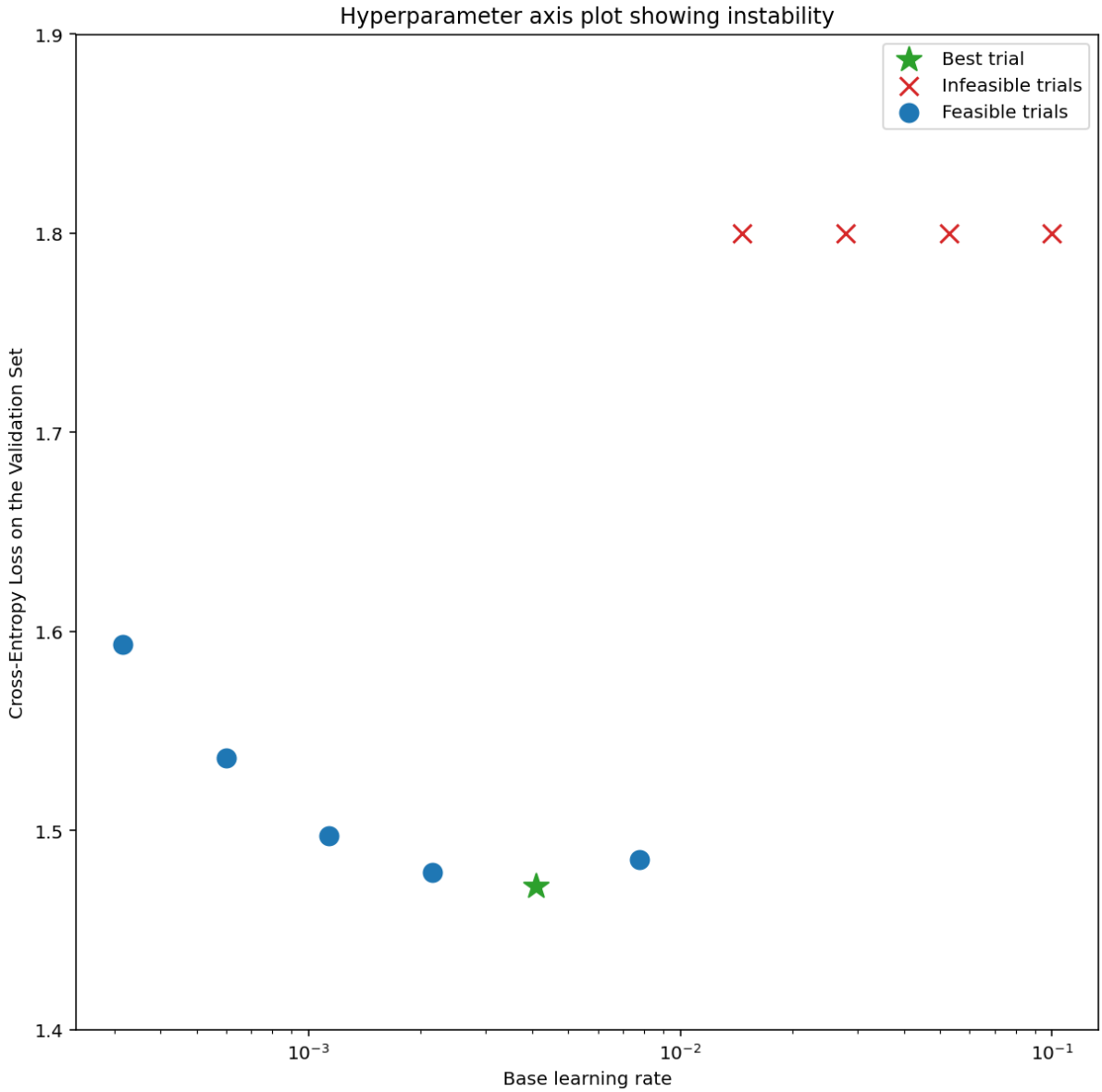
איור 4 .שינוי הצעדים בבלוק שיורי יחיד (2 (1x1 -> 2x2 בWideResnet-גורם לחוסר יציבות באימון.



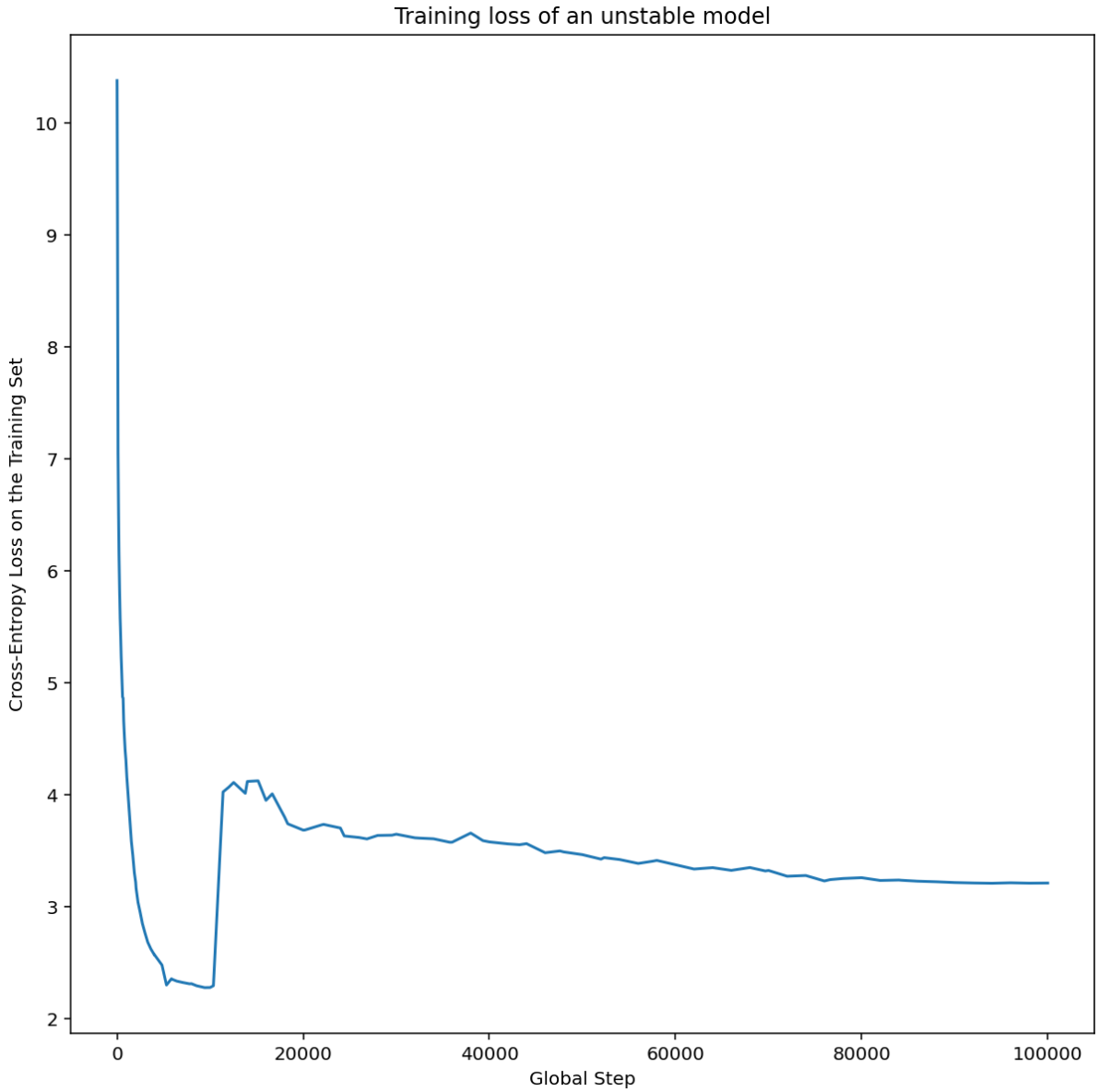
איור 5 .הערך של הערכות תכופות יותר בתחילת ההדרכה. האפשרות הזו שימושית אם אתם חושדים שהמודל סובל מחוסר יציבות בשלב מוקדם של האימון.



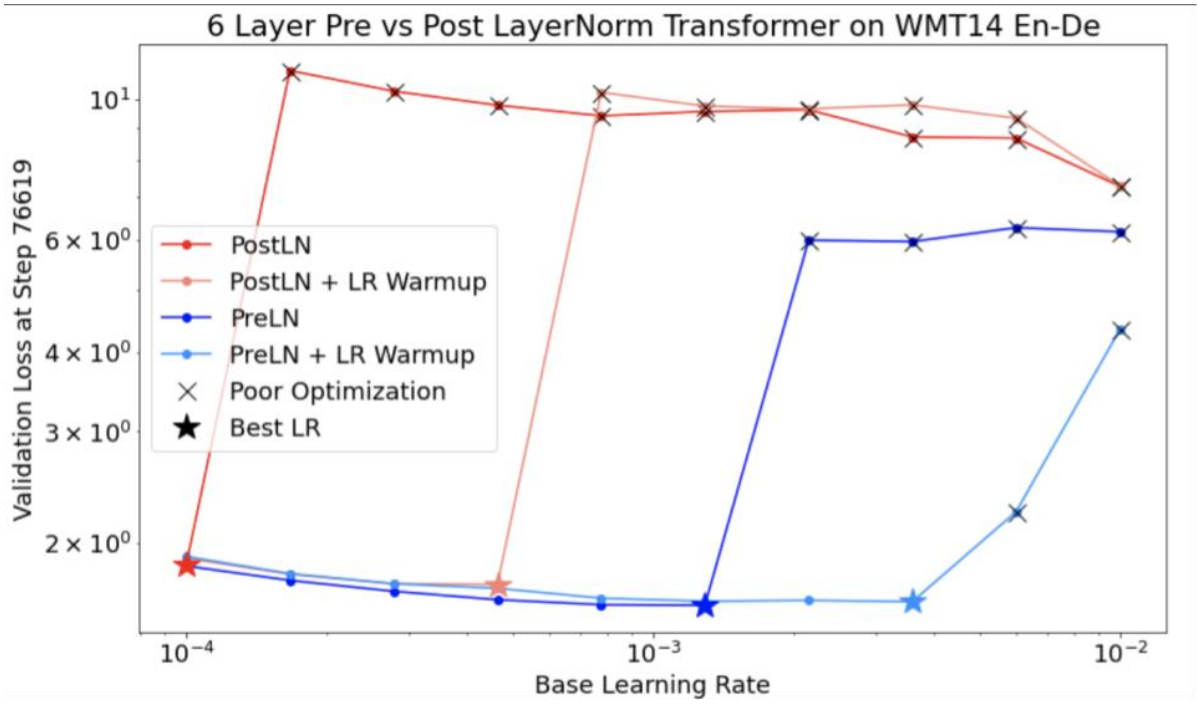
איור 6 .דוגמה לחוסר יציבות במהלך תקופת החימום (שימו לב לסולם הלוגריתמי של הציר האופקי). במקרה הזה, נדרשו 40,000 צעדים של חימום כדי שהאימון יצליח.



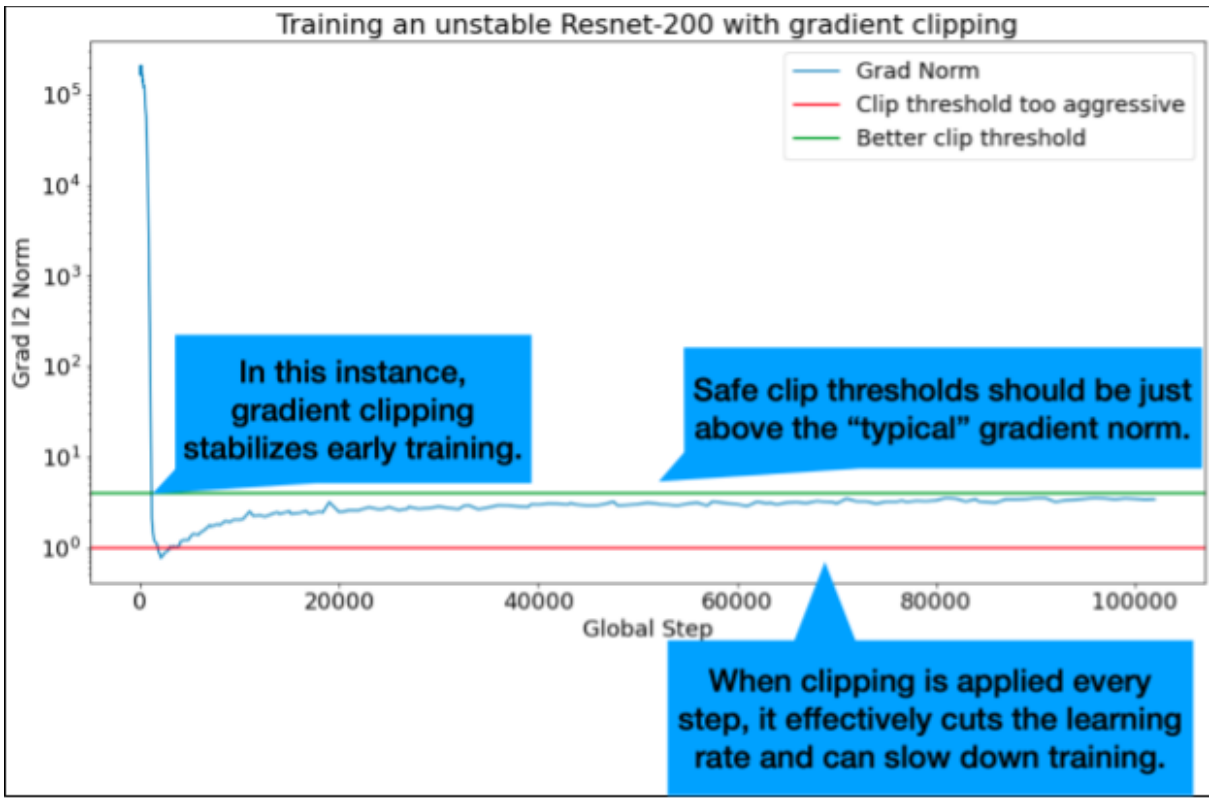
איור 7א .דוגמה לתרשים של ציר היפרפרמטרים של מודל שמציג חוסר יציבות באימון. שיעור הלמידה הטוב ביותר הוא בגבול האפשרי. ניסוי 'לא אפשרי' הוא ניסוי שמפיק ערכי NaN או ערכים גבוהים באופן לא אופייני של הפסד.



איור 7ב .מוצג איך אפשר לבדוק את זה באמצעות בחינת הפסד האימון של מודל שאומן עם קצב למידה שגדול פי 5 או פי 10 מהשיא הזה. אם בתרשים הזה רואים עלייה פתאומית בערך ה loss-אחרי ירידה יציבה)לדוגמה, בשלב 10k~ בתרשים שלמעלה ,(סביר להניח שהמודל סובל מחוסר יציבות באופטימיזציה.



איור 8 .השפעה חיובית של חימום קצב הלמידה על טיפול בחוסר יציבות באימון.



איור 9 Gradient clipping correcting early training instability ..