Neural Network For Images

29 במרץ 2023

$Ex1$
איתמר שכטר
315092759
$itamar.sh$

שאלות פרקטיות:
שאלה פרקטית **ראשונה** $Architecture$:
בחרתי להשתמש בארכיטקטורה בסיסית ולהעלות את מספר ה$channels$ בכל השכבות.
הארכטקטורה הבסיסית הייתה בנויה מ:
1) שכבת קונבלוציה כשגודל הפילטר הוא 5 על 5 ובעומק 3 כי אנחנו קיבלנו קלט של RGB. עם אקטיבציית $relu$.
למעשה בשלב הזה אנחנו לוקחים תמונה בגודל 32 על 32 על 3 והופכים אותה ל28 על 28 על מספר הערוצים שנבחר. ($padding$=$valid$)
2) שכבת $max\ pooling$ עם $stride$ 2 על 2 ובגודל 2 על 2.
בשלב זה אנחנו מצמצמים שוב את מימדי התמונה ל 14 ל 14 ומשאירים את כמות הערוצים אותו דבר.
3) שוב שכבת קונבלוציה כשגודל הפילטר הוא 5 על 5 ומספר הערוצים ככמות הערוצים שהיו בפלט השכבה הראשונה. עם אקטיבציית $relu$.
מימדי התמונה ישתנו ל10 על 10.
4) שוב שכבת $max\ pooling$ עם $stride$ 2 על 2 ובגודל 2 על 2.
בשלב זה אנחנו מצפים לקבל פלט בגודל 5 על 5 כפול מספר הערוצים בפלט השכבה הקודמת.
5) בשלב זה משטחים את התמונה לוקטור ומכניסים את הקלט לשכבת FC עם גודל פלט 10 כגודל כמות המחלקות שלנו שלהם אנחנו רוצים לבנות וקטור הסתברות. עם אקטיבציית $relu$.
- אני אעיר שיש לנו הייפר פרמטר של קצה הלמידה $learning\ rate = 0.001$, מספר האפוקים שבחרתי הוא 15.

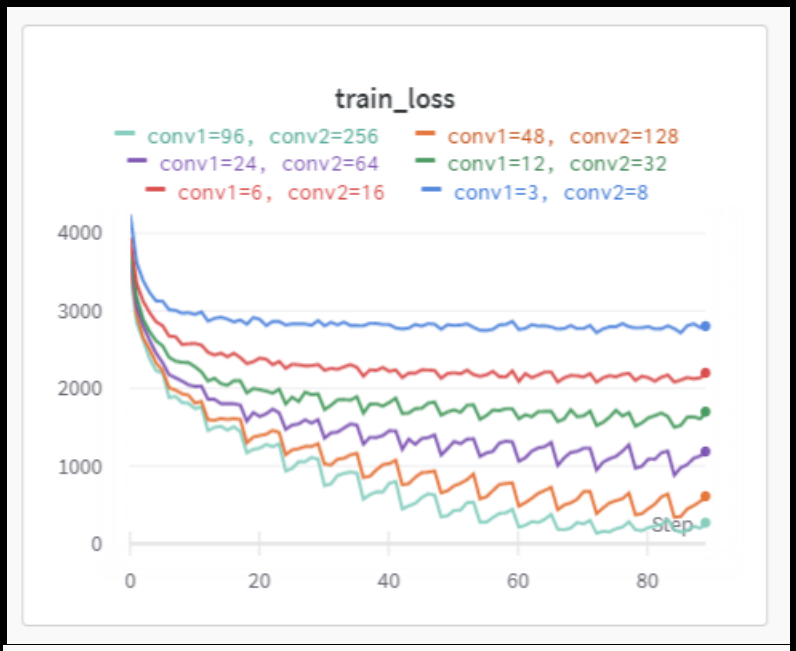
כל המשחק מעתה והלאה יהיה בכמות הערוצים שנבחר ל2 שכבות הקונבולוציה.
בכוונה אנחנו שומרים על פתרון יחסית פשוט, בלי הרבה שכבות קונבלוציה קטנות בגודל 3 על 3 כי חשבתי שהרבה שכבות קטנות יהיה קשה לבחור היפר פרמטרים שונים ולהסיק מסקנות בצורה ברורה. למרות שהבנתי שרוב הרשתות הקיימות בסוף מתכנסות לשכבות 3 על 3 כי זה נותן יותר $perceptive\ field$ בפחות כוח עיבוד.

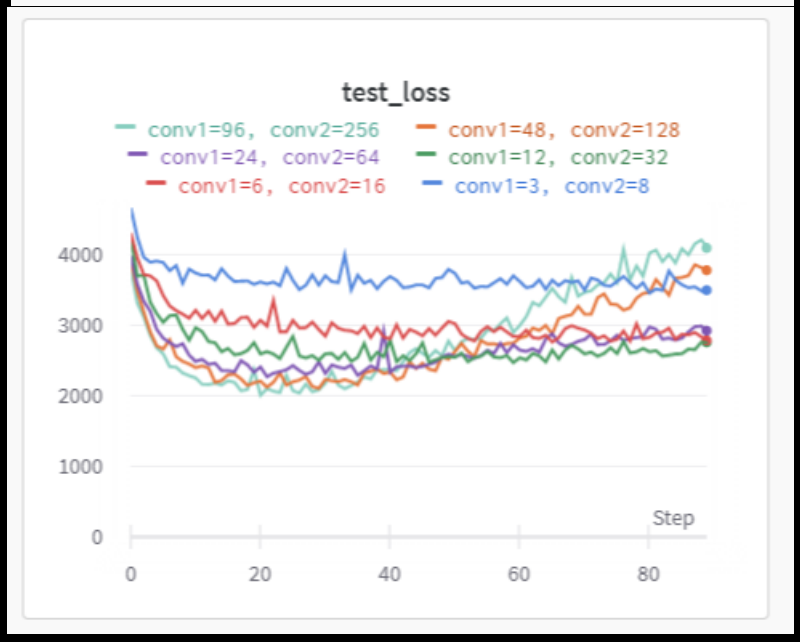
אני לקחתי מערכת עם מעט מאוד פרמטרים.

על כמות הפרמטרים אני שולט בכמות הערוצים בשכבות הקונבולוציה.
בריצה הראשונה התחלתי עם:
3 ערוצי פלט בשכבת קונבולוציה הראשונה.
8 ערוצי פלט בשכבת הקובלוציה השנייה.
נחשב את כמות הפרמטרים:
$conv1$: גודל כל קרנל הוא 5 על 5 על 3 ולכן גודל קרנל הוא 75 ונוסיף לזה באייס אז 76. יש לנו 3 ערוצים פלט ולכן $76 * 3 = 228$.
$conv2$: גודל כל קרנל הוא 5 על 5 על 3 ולכן שוב נקבל 76 לקרנל יחיד. יש לנו 8 ערוצי פלט ולכן 8 קרנלים, אז החישוב הוא: $76 * 8 = 608$.
FC: נבחין שגודל הקלט הוא גודל התמונה שזה 5 על 5 בשלב הזה (ראו סעיף 4 למעלה) והעומק הוא לפי ערוצי הפלט של השכבה השנייה שזה 8.
ולכן נקבל שיש לנו סה"כ וקטור קלט באורך $5 * 5 * 8 = 200$. הפלט שלנו הוא בגודל 10 ולכן כמות הפרמטרים היא מטריצה בגודל $200 * 10$ ועוד באייס באורך הפלט שהוא 10 ולכן סה"כ $201 * 10 = 2010$.
סה"כ: $228 + 608 + 2010 = 2846$.

בכל ריצה מהריצות הבאות הכפלתי ב2 את כמות הערוצים מהריצה שלפניה בקונבלוציה, בעיקר כדי להגיע יחסית מהר ל$overfit$.
-
מדדתי בכל 2000 מיני באצ'ים את ה$train\ loss$, זה פחות מדויק כי יש פה רנדומליות אבל בסה"כ אנחנו מצפים לראות גרף של ירידה.
אם לא נראה גרף של ירידה ב$train$ סימן שיש לנו $underfit$.
בנוסף מדדתי את ה$test\,loss$ על פני כל ה$test\ data$ בכל 2000 מיני באצ'ים כדי לראות ירידה גם פה. במידה ונתחיל לראות נקודה בה ה$test$ מתחיל לעלות נדע שהגענו ל$overfit$.

אלה התוצאות שקיבלתי:





אני אסביר:

כל צבע כאן מציג אחת מ6 הרשתות שאימנתי, השם שלהן הוא לפי כמות הצ'אנלים שיש בפלט שכבת הקונבולוציה הראשונה ופלט שכבת הקונבולוציה השנייה. (יש רק שני שכבות)

לדוגמא $conv1 = 24, conv2 = 64$ מציג רשת עם שכבת קונבלוציה ראשונה בעלת 24 קרנלים ופולטת קלט עם 24 ערוצים. כלומר אם הקלט הוא הוא תמונות בגודל $(32, 32, 3)$ אז הפלט הוא טנזורים בגודל: $(28, 28, 24)$. כי כל השכבות הן עם קרנל 5 על 5. והשכבה

השנייה (אחרי $maxpool$) תקבל קלט $(14,14,24)$ ותחזיר $(10,10,64)$.

בגרף הלמידה ניתן לראות שהצבע הכחול ($conv1 = 3, conv2 = 8$) בעל הכי פחות פרמטרים ללמידה הוא כמעט ולא לומד ואכן גם ה$test\ loss$ שלו די נשאר במקום.

הגרף האדום, ($conv1 = 6, conv2 = 16$), עם כפול ערוצים בכל שכבת קונבלוציה וגם בעקיפין כפול פרמטרים נלמדים בשכבת הFC, מצליח כבר ליצור גרף למידה יורד, אך לא בהצלחה גבוהה.

בגרף הירוק ($conv1 = 12, conv2 = 32$), אנחנו כבר מתחילים להתייצב ולקבל תוצאה דומה לאדום בטסט למרות שאנחנו עם כפול פרמטרים ועם לוס נמוך יותר באימון.

הגרף הסגול הראשון ($conv1 = 24, conv2 = 64$), שמתחיל להראות $overfit$ אבל מאוד קטן כשהלוס שלו בטסט קצת עולה מעל קודמיו בסוף.

הכתום ($conv1 = 24, conv2 = 64$), מגיע כבר ללוס די גדול. עוקף אפילו את הכחול שכמעט ולא לומד.

והתכלת, שהוא כבר עם כמות נכבדת של פרמטרים, מצליח לשנן את הדאטא די ברצינות ולהגיע לתוצאות גרועות אפילו יותר בטסט לוס כשהאימון שלו ממש מתחת לכולם.

מסקנה כרגע שהפתרון האידיאלי מהשישה הוא הגרף הירוק ($conv1 = 12, conv2 = 32$), כי הוא היחיד שלא מגיע ל$overfit$.

לצרוך סגירת מעגל נחשב זריז את הפרמטרים שלו:

$conv1$: גודל כל קרנל הוא 5 על 5 על 3 שזה 75 ועוד באייס שזה 76. יש לנו 12 ערוצים כאלה ולכן: 912.

$conv2$: גודל כל קרנל הוא 5 על 5 על 12 שזה 300 ועוד 1 זה 301. יש לנו 32 ערוצים כאלה ולכן: 9632.

fc: יש לנו קלט בגודל 5 על 5 על 32 שכשנשטח אותו נקבל ווקטור באורך 800. הפלט שלנו הוא ווקטור באורך 10 ולכן צריך מטריצה באורך 800 על 10 אז 8000 ויש באייס של 10 ולכן 8010.

נסכום הכל ונקבל 8010+9632+912=11354 פרמטרים נלמדים.

שאלה פרקטית שנייה $Importance\ of\ \ Non-Linearity$**:**

התבקשתי למחוק את הפעולות הלא לינאריות שברשת.

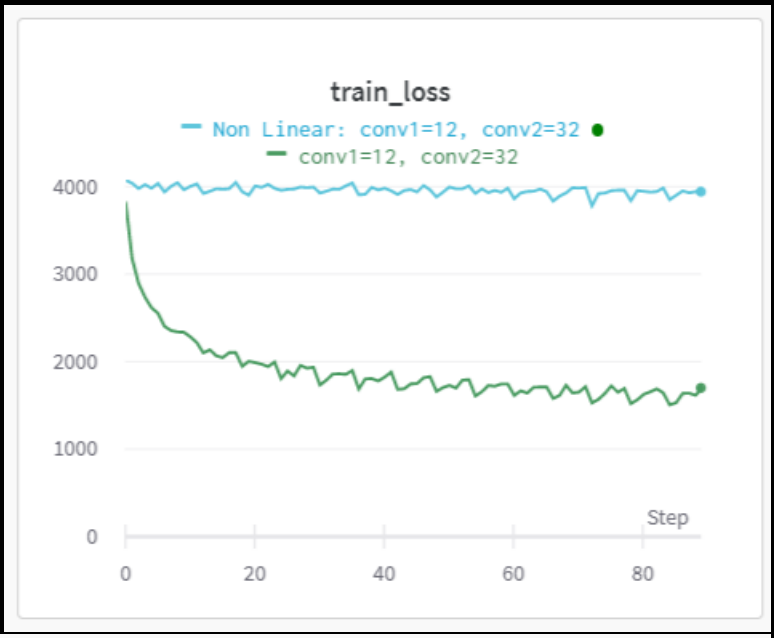
אז את האקטיבציות של ה$relu$ זה קל כי לא זה לא הורס את המימדים.

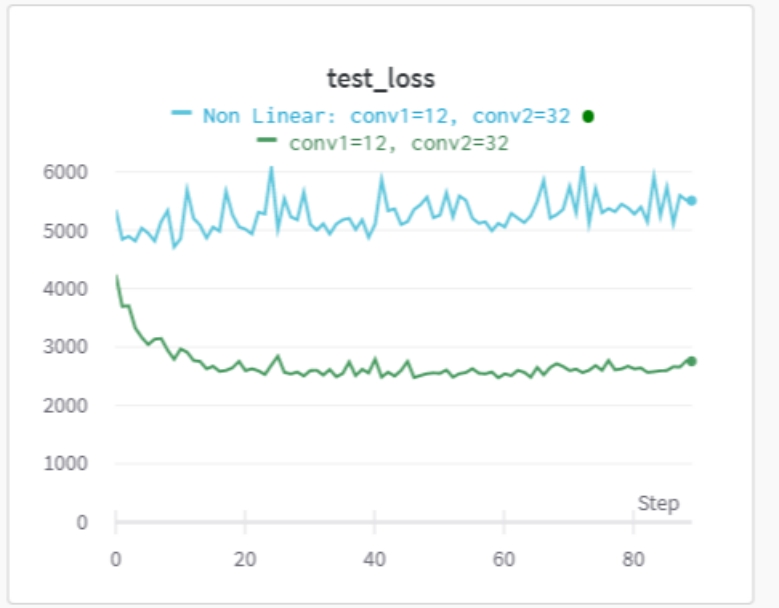
כשנוריד את ה$max\ pooling$ת שזה גם לא לינארית אנחנו נקבל קלט לשכבת הfc שהיא בגודל 24 על 24 על 32 וזה 18,432 פרמטרים נלמדים רק לשכבת הfc.

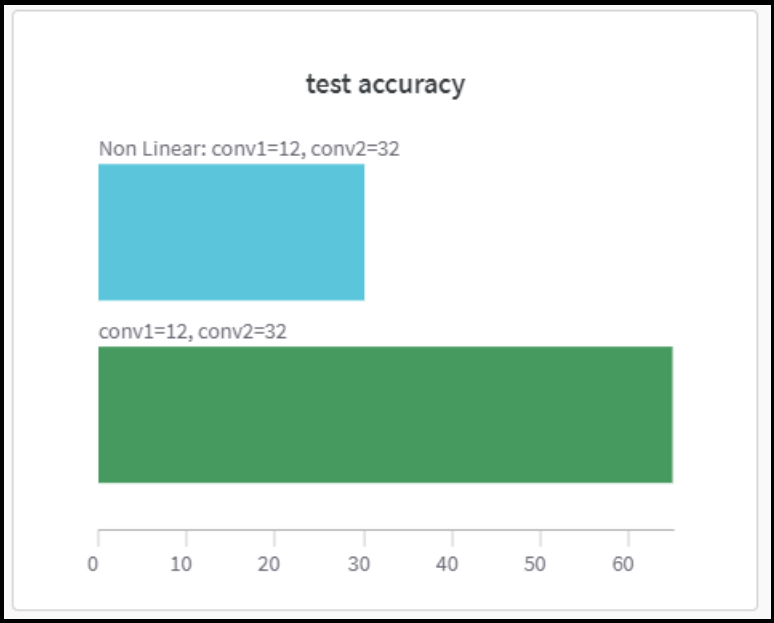
זה מכיוון שלא צימצמנו את המימדים של הרשת.

לסיכום נבדוק עכשיו רשת עם 2 שכבות קונבולוציה בעלות קרנל 5 על 5 כשלראשונה פלט בעומק 12 ולשנייה פלט בעומק 32 והתוצאה תיכנס לfc פשוט שמחזיר וקטור באורך 10.

התוצאות של הרשת הזו הן (בתכלת התוצאות כשמורידים כל רכיב לא לינארי וירוק זו הרשת המקורית):


train_loss
Non Linear: conv1=12, conv2=32
conv1=12, conv2=32
4000
3000
2000
1000
0
Step
0
20
40
60
80


test_loss
Non Linear: conv1=12, conv2=32
conv1=12, conv2=32
6000
5000
4000
3000
2000
1000
0
Step
0
20
40
60
80

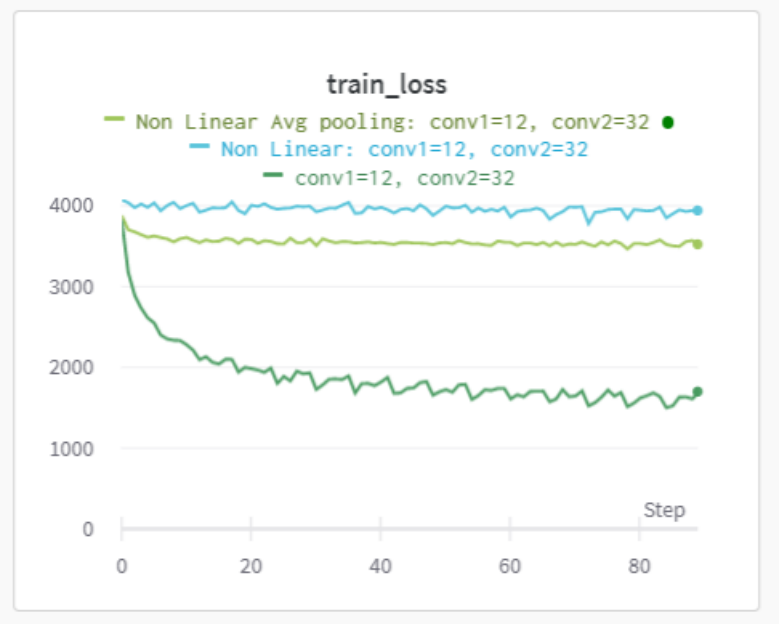


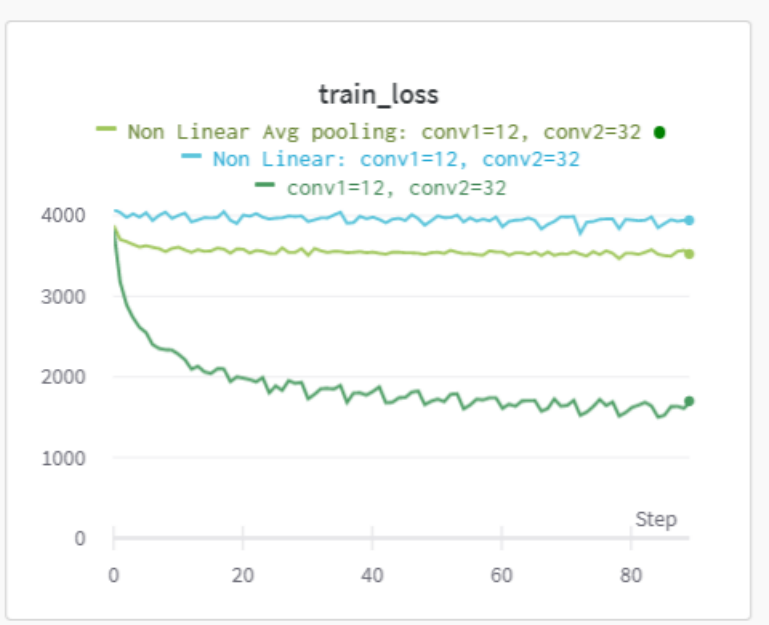
ניתן לראות שאפילו ה$train\ loss$ לא יורד וגם ה$test\ loss$ לא יציב ובטח שלא יורד והדיוק בהתאם, 30 אחוז, לעומת 65 במודל שברחנו מהסעיף הקודם.

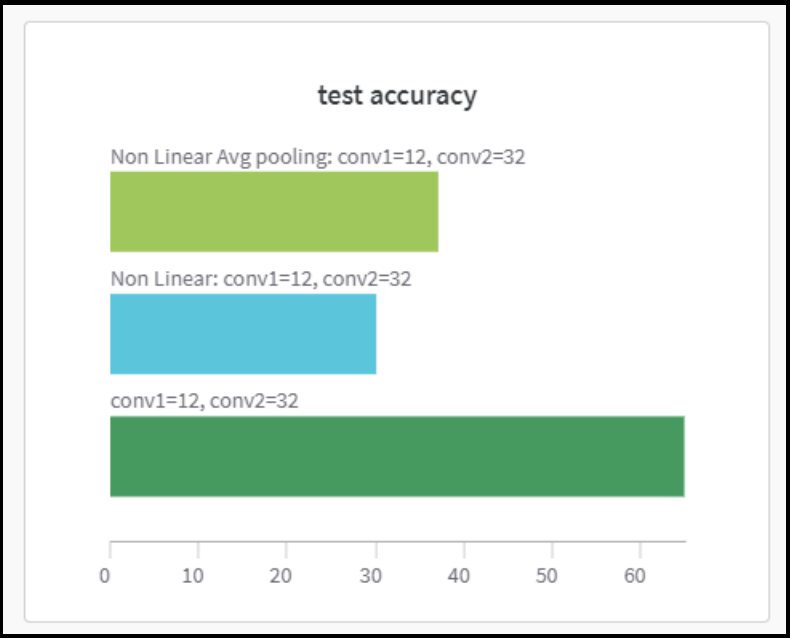
בואו ננסה, לפני שנעביר לסעיף הבא לאפשר לרשת להיראות דומה יותר בכמות הפרמטרים למודל הירוק, כי יותר פרמטרים זה משימה אחרת שאולי לא מצליחה להתכנס פשוט.

במקרה זה נמיר את שכבות ה$max\ pooling$ לשכבות $average\ pooling$ וככה נישאר על אותה כמות פרמטרים בלי לינאריות.

אלו התוצאות: (הטורקיז זה הגרף החדש כתוספת על הגרפים הקודמים)


train_loss
Non Linear Avg pooling: conv1=12, conv2=32
Non Linear: conv1=12, conv2=32
conv1=12, conv2=32
4000
3000
2000
1000
0
Step
0
20
40
60
80


train_loss
Non Linear Avg pooling: conv1=12, conv2=32
Non Linear: conv1=12, conv2=32
conv1=12, conv2=32
4000
3000
2000
1000
0
Step
0
20
40
60
80



(באחוזים)

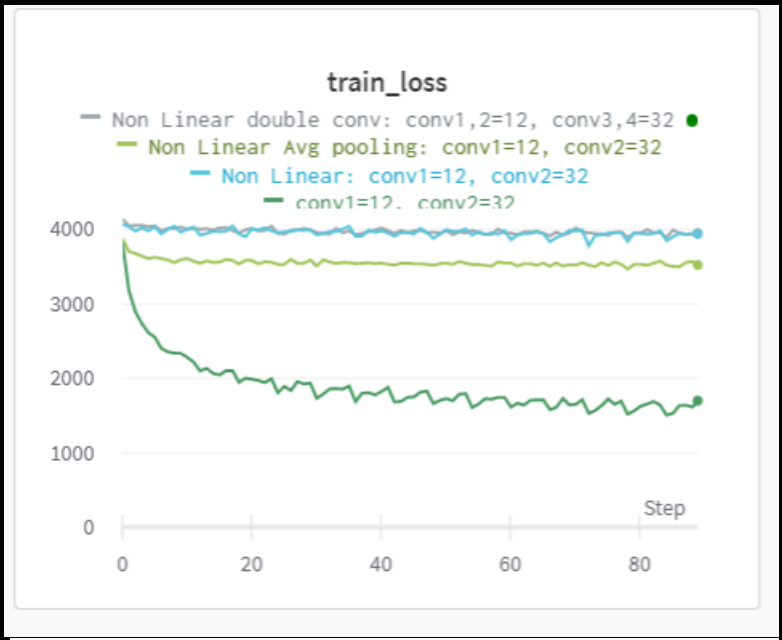
אפשר לראות שאפילו עכשיו, כשהשארנו את השינויים בגודל הקלט והפלט של כל שכבה באמצעות הוספת $avgPool$ עדיין התוצאות משמעותיות שונות כשיש לינאריות.

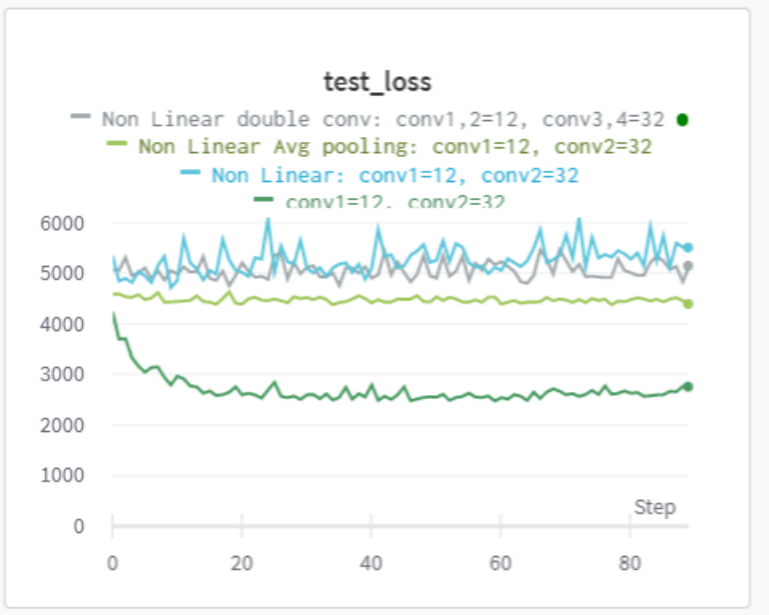
לסיום נחזור לפתרון ללא ה$AvgPool$ רק שהפעם יהיו לנו הרבה יותר שכבות קונבולוציה של 5 על 5.

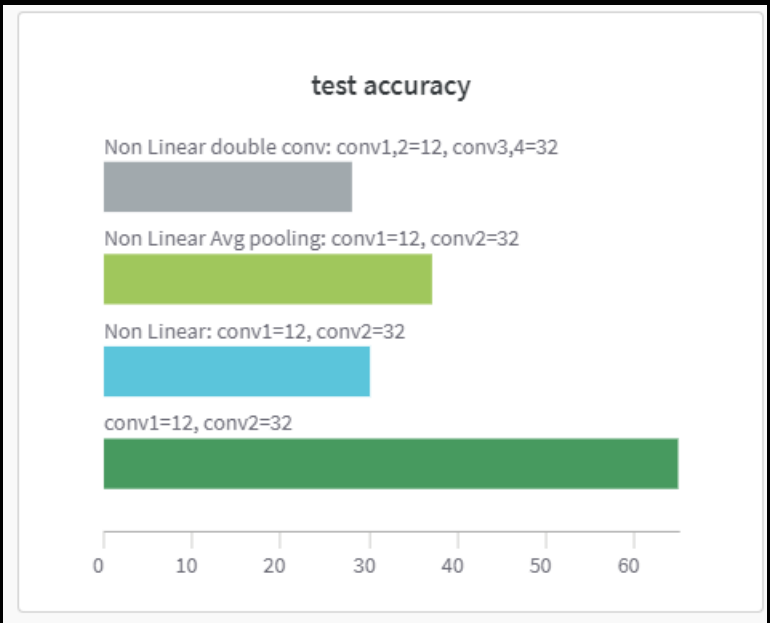
אם עד עכשיו היו לנו 2 שכבות, הפעם נוסיף 2 שכבות ונקבל ארבע שכבות קונבולוציה. 2 ראשונות יהיו עם 12 ערוצים והאחרונות עם 32 ערוצים.

נרצה לקבל תוצאות דומות לגרף התכלת, כי כשאין לינאריות אז הכל זה כפל מטריצות אחד גדול. (גילוי נאות - שורה זו נכתבה לפני הרצת המודל ואני שמח שזה אכן מה שקרה)

הנה התוצאות: (באפור)


train_loss
Non Linear double conv: conv1,2=12, conv3,4=32
Non Linear Avg pooling: conv1=12, conv2=32
Non Linear: conv1=12, conv2=32
conv1=12, conv2=32
4000
3000
2000
1000
0
Step
0
20
40
60
80


test_loss
Non Linear double conv: conv1,2=12, conv3,4=32
Non Linear Avg pooling: conv1=12, conv2=32
Non Linear: conv1=12, conv2=32
conv1=12, conv2=32
6000
5000
4000
3000
2000
1000
0
Step
0
20
40
60
80



אפשר לראות בגרפים כמה מתואמים הגרפים האפורים והטורקיז, כשההבדל ביניהם היא מכפלה של כמות הפרמטרים הנלמדים בשכבות הקונבולוציה!

כשהיה לנו לינאריות אפילו הזזה של פחות פרמטרים שינתה והפרידה בין הגרפים, במיוחד על סט האימון.

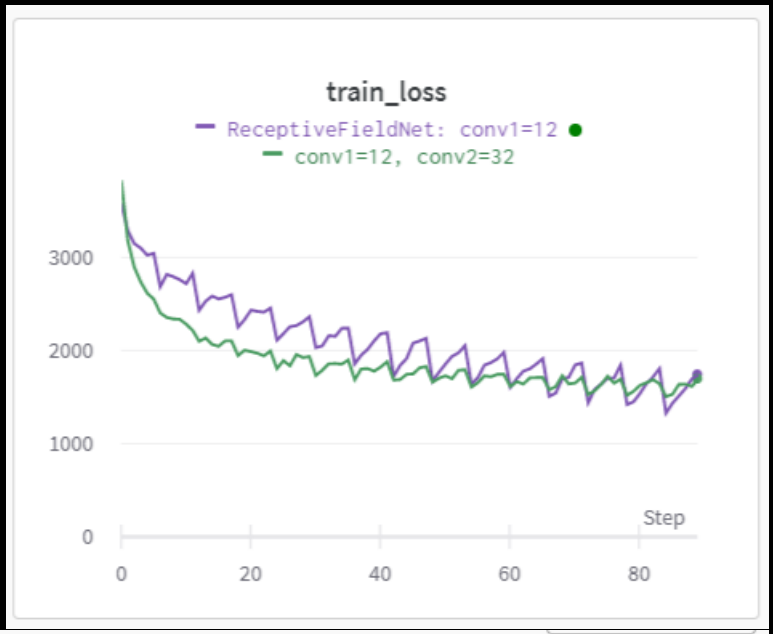
כמו שכבר אמרתי, מאחר ויש לינאריות בכל הפרמטרים, אנחנו מקבלים תלות בין הפרמטרים ככה שהכל מסתכם למטריצה אחת באמצעות הכפלת מטריצות במקום למידה של מטריצות בלתי תלויות זו בזו.

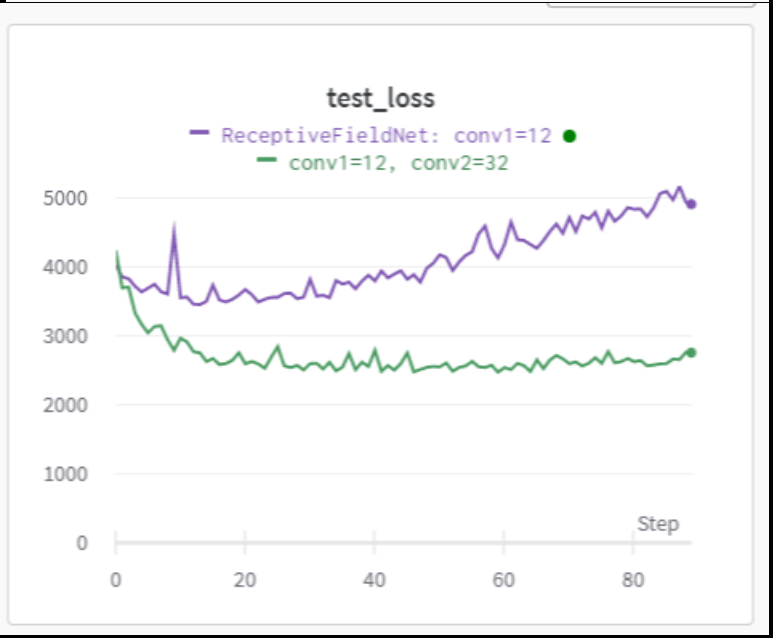
שאלה פרקטית שלישית $Cascaded\ Receptive\ Field$.**:**

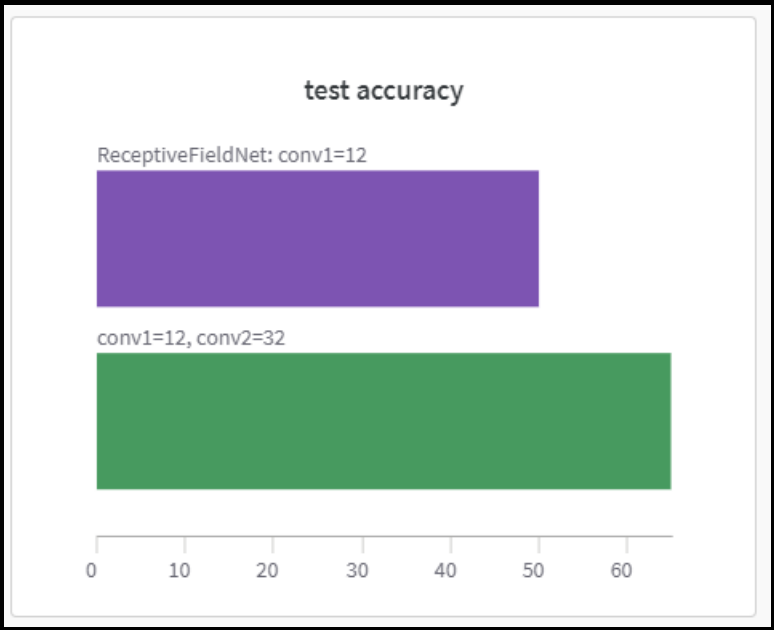
בחרתי באפשרות הראשונה ליישום, בעיקר כי היא נראתה לי כרשת שמציגה בקלות יותר את מטרת הסעיף - להראות את הכוח של ה$receptive\ field$.

אז עכשיו יהיה לנו רשת עם שכבת קונבולוציה אחת עם קרנל 5 על 5 ו12 ערוצים שאת הפלט שלה נכניס לשכבת הfc שלנו שתחזיר פלט באורך 10.

הנה התוצאות בסגול ביחס לתוצאות המקוריות בירוק:


train_loss
ReceptiveFieldNet: conv1=12
conv1=12, conv2=32
3000
2000
1000
0
Step
0
20
40
60
80


test_loss
ReceptiveFieldNet: conv1=12
conv1=12, conv2=32
5000
4000
3000
2000
1000
0
Step
0
20
40
60
80



נבחין בשגיאה הגבוהה ב$test\ loss$ שאמור להעיד על $overfit$ כי השגיאה של ה$train$ כן ירדה, כנראה כי לא היה הרבה פרמטרים ללמוד.

מצד שני הדיוק הוא 50 אחוז, יחסית גרוע, יותר טוב מהטלת מטבע כי יש לנו 10 מחלקות אבל עדיין ראינו שהרבה מהמודלים הפשוטים שלנו הצליחו לעקוף את זה.

נראה שהבעיה היא לא במספר הפרמטרים ללמידה:

בשכבת ה$conv1$: היו לנו קרנלים בגודל 5 על 5 על 3 שזה 75 עם באייס 76 פרמטרים לכל קרנל כשיש לנו 12 קרנלים אז 12*76=912

ובשכבה fc: מקבלים קלט בגודל 12*28*28=9408 ופלט בגודל 10 ולכן עם הבאייס: $9409 * 10 = 94090$.

שזה כמות רצינית של פרמטרים ללמידה. ולכן הבעיה כאן היא שלא הצלחנו בכלל לתפוס כמו שצריך את התמונה ולכן הלמידה לא הועילה.

שאלות תיאורטיות:

1) נסביר את השאלה: אנחנו רוצים להוכיח שאם מתקיים:

$$L[x(i+k)]_{(j)} = L[x(i)]_{(j+k)}$$

כשL הוא אופרטור לינארי, x הוא וקטור קלט, $x(i)$ זה הקואורדינטה הi בוקטור הקלט, k הוא סקלר שיסמן לנו אופסט מסוים על וקטור הקלט.

$L[x(i)]$ מחזיר לנו וקטור פלט מהפעולה L על סקלר כלשהו. L היא הכפלה במטריצה ולכן לכל קואורדינטה בוקטור הקלט אנחנו מקבלים וקטור כפלט.

הפעולה $L[x(i)]_{(j)}$ מסמנת את הקואורדינטה הj בוקטור הפלט.

עלינו להוכיח שמדובר פה בפעולת קונבולוציה.

רמז: נפרק את $x(i)$ לסכום ממושקל של פונקציות δ ואז להשתמש בלינאריות של L.

שאלה מכווינה: איזה סיגנל כקלט ייתן לנו פעולה דומה לקונבולוציה?

פתרון: נראה שמדובר באופרטור קונבולוציה.
נבחין כי:

$$x(i) = \sum_{k=1}^{n} x(k) \cdot \delta(i-k)$$

מלינאריות L נקבל כי:

$$L[x(i)]_{(j)} = L\left[\sum_{k=1}^{n} x(k) \cdot \delta(i-k)\right]_{(j)} = \sum_{k=1}^{n} \left[x(k) \cdot L[\delta(i-k)]_{(j)}\right]$$

$$= \sum_{k=1}^{n} \left[x(k) \cdot L[\delta(i)]_{(j-k)}\right] = \sum_{k=1}^{n} \left[x(k) \cdot l(j-k)\right] = (l * x)_j$$

2) אין משמעות לסדר, מכיוון שאנחנו הולכים להכפיל את הוקטור תוצאה במטריצה שאותה נלמד, אז כל ערך יחיד i בוקטור הולך להיות מוכפל בשורה ייעודית i במטריצה ולכן אנחנו פשוט נלמד את השורה הזאת להתאים לערכים שמגיעים לקואורדינטה הi בקלט. ואם היינו מחליפים את i וj, מכיוון שאין תלות בין המשקולות ברשת אז היינו מקבלים שהשורות המתאימות היו מתחלפות במטריצת המשקולות פשוט. (הערה: הבנתי שיש מקרים בהם עבודה עם $batch$ מייצרת איזשהי תלות בסדר ברמה הפרקטית ולא התיאורטית, אבל אני לא מכיר את זה מספיק ולכן אני משאיר את התשובה שלי כמו שהיא).

3) א) לא, $Relu$ אינה LTI מכיוון שהיא אינה לינארית. אפשר לקחת דוגמא קטנה של הזזה של ערך יחיד 1 בקלט במינוס 2 ולראות שהערך יגיע ל0 ואם נזיז אחרי ההפעלה נגיע ל1-

ב) לא, $strided\ pooling$ אמנם פעולה לינארית ואפשר להציג אותה כמטריצה מתאימה עם עמודות 1 ו0. אך אם נבצע $translation$ על הקלט הפלט יכול לקבל שינוי חד בערכים לפי המיקום של הערכים במיקומים המתאימים.

ג) לא, מכיוון שחיבור בקבוע אינו לינארי. אפשר לקחת וקטור פשוט לבצע הזזה עליו ולראות שהסכום החדש ייראה שונה לחלוטין.

ד) לא, אמנם יש לנו כאן הכפלה במטריצה אבל בהנחה ונזיז את וקטור קלט אז הערכים יוכפלו בעמודות שונות שיכולות להיות עם ערכים שונים זה מזה כך שהפלט ייצא שונה.