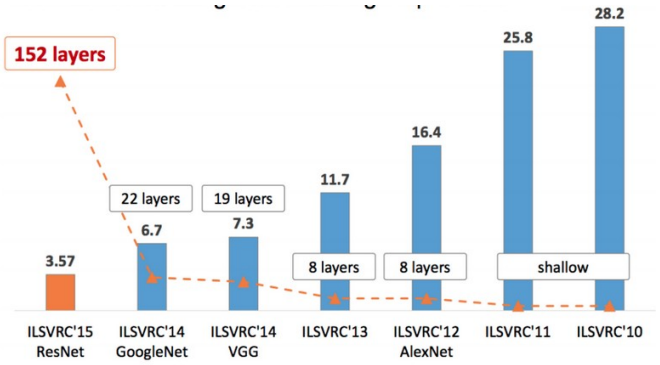
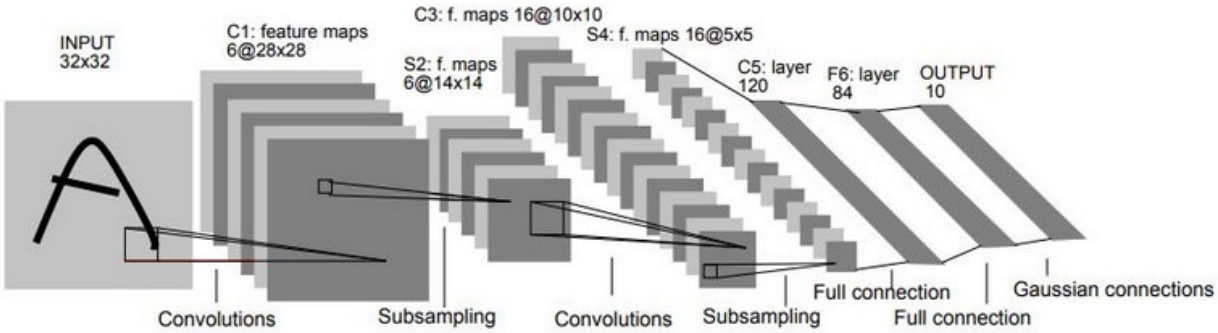
CNN Architectures (2015)

נפרט על שלבי הפיתוח והארכיטקטורות CNN השונות שזכו באתגר של ImageNet לזיהוי אובייקטים ILSVRC.



LeNet-5

המטרה של LeNet-5 היא זיהוי של ספרות בכתב יד. מתחילים עם תמונה בגודל 32X32X1 והמטרה שלנו היא זיהוי הספרה. מבנה הרשת:



convo → pool → convo → pool → FC → FC → softmax

הארכיטקטורה של LeNet-5 מורכבת משתי שכבות קונבולוציה(no padding) ו average pooling ושכבת אקטיבציה Tanh לאחר כל שכבת קונבולוציה, ואחריהן שתי שכבות fully connected ולבסוף מסווג softnax.

תחילה, בשכבה הראשונה אנו משתמשים ב 6 פילטרים בגודל של 5X5 ו stride 1 הפלט הוא 28X28X6. לאחר מכן אנו מפעילים את pooling 2X2 הפלט הוא 14X14X6. לאחר מכן הפלט עובר לשכבת קונבולוציה נוספת הכוללת 16 פילטרים בגודל 5X5 הפלט הוא 10X10X16. לאחר מכן שכבת pooling 2X2 נוספת שהפלט שלה הוא 5X5X16. לאחר מכן שתי שכבות fully connected הראשונה כוללת 120, השניה 84 נוירוניים, המחוברים (בגרסה המודרנית) לשכבת softmax בעלת פלט של 10 הסתברויות עבור כל ספרה.

בארכיטקטורה זאת אנו רואים כי ככל שאנו מעמיקים ברשת הגובה והרוחב יורדים ומספר ה channels עולה.

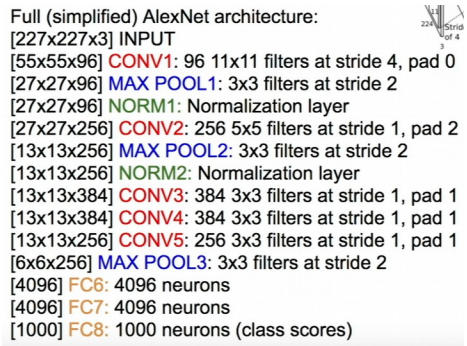
אז כמה משקולות יש לנו במודל הזה?

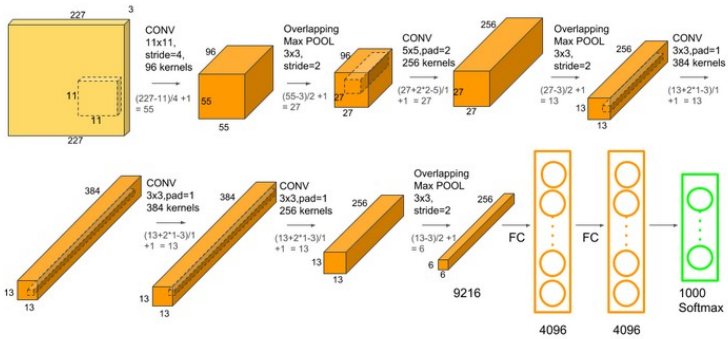
(5*5+ 1)*6 + (6*5*5+ 1)*16 + (5*5*16+ 1)*120 + (120+ 1)*84+ (84*10+ 10) = 61,706

כלומר זוהי רשת קטנה יחסית לרשתות של ימינו.

AlexNet

AlexNet התפרסם בזכות זכייה בתחרות ImageNet ILSVRC –2012 בשנת 2012 בפער גדול ממתחריו.





בתחרות ILSVRC 2012 התבקשו המתחרים לפתור בעיית סיווג תמונה בין 1000 אפשרויות שונות (כלב, חתול, וכדומה).

הקלט של הרשת היא תמונה בגודל של 227X227X3. בשכבת הקונבולוציה הראשונה 96 פילטרים בגודל של 11X11 עם stride= 4, padding= valid, ולכן פלט השכבה הראשונה 55X55X96. לאחר מכן max polling בגודל 3X3 עם stride= 2, ולכן נקבל משכבה זו פלט של 27X27X96. פלט זה עובד לשכבת קונבולוציה נוספת בעלת 256 פילטרים כל אחד בגודל של 5X5, padding= same, ולכן הפלט הוא 27X27X256, לאחר מכן על פלט זה נפעיל שוב maxpooling בגודל 3X3 עם stride= 2 ולכן הפלט הוא 13X13X256. פלט זה עובר לשתי שכבות קונבולוציה נוספות כל אחת בעלת 384 קרנלים בגודל 3X3 ו– padding= same ולכן הפלט של שכבות אלו הוא 13X13X384. פלט זה מועב לשכבת קונבולוציה נוספת בעלת 256 פילטרים בגודל של 3X3 עם פאדינג, לכן הפלט של שכבה זו הוא 13X13X256 פעל זה מועבר לשכבת max–polling 3X3 עם stride= 2 הפלט הוא 6X6X256.

פלט זה חובר לשכבת fully connected שמחוברת לעוד שכבה כזאת כאשר בכל שכבה יש 4096 נוירונים. השכבה השניה של ה fully connected מחוברת לשכבת softmax שהפלט שלה כאמור הוא וקטור בגודל 1000 המכיל את ההסתברות של כל סייווג. לשם שבירת לינאריות לאחר כל שכבת קונבולוציה ו FC מופעלת ReLU.

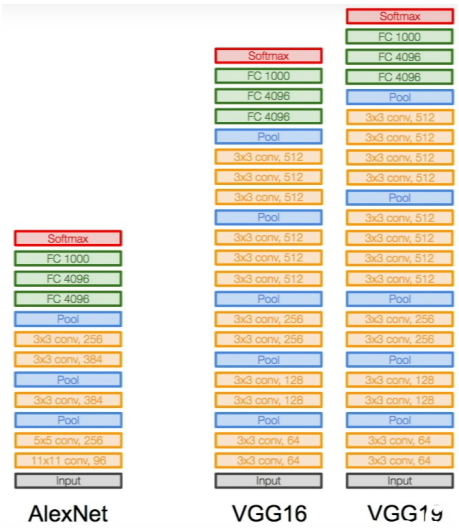
נקודות מפתח:

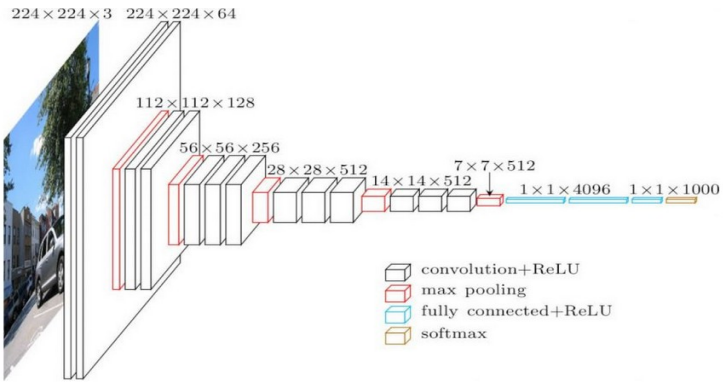
1. שימוש ב ReLU במקום ב Tanh בתור פונקציית שבירת לינאריות. מעניק רשת מהירה יותר בעלת דיוק זהה.
2. שימוש ב dropout על מנת להתמודד עם overfitting.
3. שימוש ב data augmentation.
4. שימוש ב overlap pooling על מנת להפחית את גודל הרשת.
5. Batch size= 128
6. SGD Momentum 0.9

כמה פרמטרים ברשת זו?

(3*11*11*96+ 96)+ (96*256*5*5+ 256)+ (256*384*3*3+ 384)*2+ (384*3*3*256+ 256)
)+ (6*6*256*4096+ 4096)+ (4096*4096+ 4096)+ (4096*1000+ 1000)= 61,910,632

VGG16





VGG-16 הוא אחד הארכיטקטורות המפורסמות ביותר שהוגשו ל ILSVRC-2014. הוא מניב תוצאות טובות יותר ומשפר את מודל AlexNet על ידי החלפת קרנלים גדולים (בגודל של 11 ו 5) בקרנלים בגודל 3X3. הדבר המדהים ביותר ב VGG-16 הוא הפשטות שלו. ברשת שכבות קונבולוציה עם פילטרים בגודל 3X3 בעלות שימוש קבוע ב padding same, ו stride= 1. כל שכבות ה max pooling הן בגודל 2X2 עם stride= 2, לכן לאחר כל pooling המימדים של האורך והרוחב מתכווצים בפקטור של 2.

שאלה: למה להשתמש בפילטרים קטנים יותר(3X3) ?

תשובה: לערום שלושה פילטרים של 3X3 יש השפעה דומה ל receptive field של פילטר אחד 7X7, רק עם פחות פרמטרים.

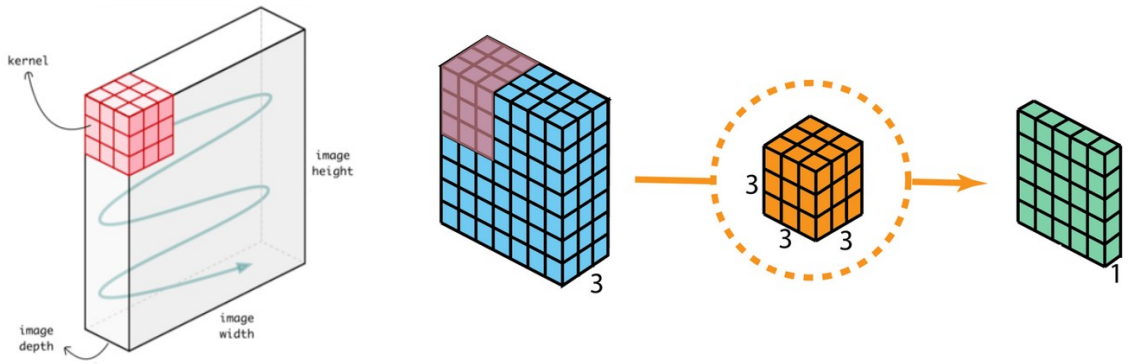
כמה פרמטרים אנו מאמנים ברשת זו ?

(3*3*3*64+ 64) +
(3*3*64*64+ 64)+ (3*3*64*128+ 128)+ (3*3*128*128+ 128)+ (3*3*128*256+ 256)+ (
3*3*256*256)*2+ (2*2*256*512+ 512)+ (3*3*512*512+ 512)*2+ (3*3*512*512+ 512)
*3+ (7*7*512*4096+ 4096)+ (4096*4096+ 4096)+ (4096*1000+ 1000)

1X1 Convolution layer, Network In Network

במבט ראשון זה נראה חסר טעם לבצע קונבולוציה עם ספרה אחת. עם זאת, 1X1convolution הוכח ככלי שימושי ביותר וכאשר הוא מתופעל בצורה נכונה, הוא מועיל ביצירת רשתות עמוקות.

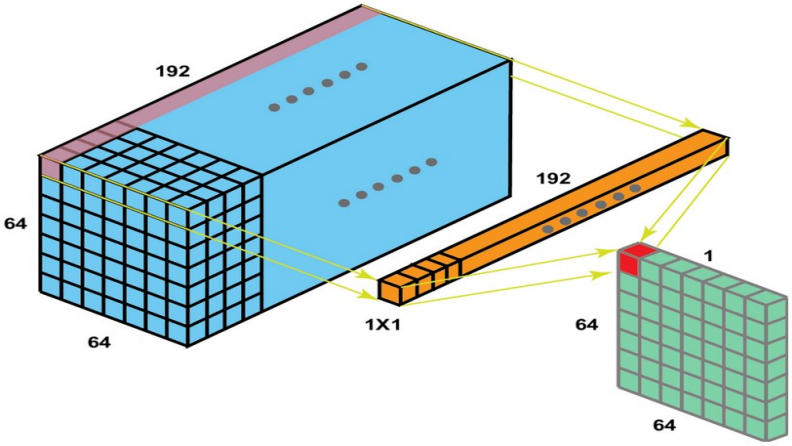
- מטריצת הקלט עבור שכבת קונבולוציה, ברוב המקרים מכילה יותר מערוץ אחד. מספר ערוצי הקלט מכונה עומק המטריצה.
  לדוגמה: שכבת קונבולוציה המקבלת תמונת RGB בגודל 102X102 תכיל 3 ערוצי כניסה, כך שהכניסה היא 102X102X3.
- לכל קרנל יש עומק זהה לעומק של מטריצת הקלט (פרט למקרים מיוחדים).
  לדוגמה: עבור פילטר בגודל 3X3 העובר על קלט בגודל של 3 ערוצים יהיו גם שלושה ערוצים. ולכן גודלו הוא 3X3X3.
- העומק של הפלט של שכבת קונבולוציה הוא מספר הפילטרים שיש באותו שכבה.
  לדוגמה: עבור קלט לשכבת קונבולוציה בגודל 102X102X3 עם פילטר יחיד בגודל 3X3X3 עומק הפלט יהיה 1 בלבד.



1X1 convolution הוצג לראשונה במאמר של Min Lin שכבת קונבולוציה 1X1 שמשה לשם 'Cross Channel Down sampling' or Cross Channel Pooling. במילים אחרות, קונבולוציית 1X1 שימשה לשם הפחתת נספר הערוצים תוך הצגת אי-ליניאריות.קונבולוציה 1X1 זה פשוט פילטר בגודל 1X1.

לדוגמה" עבור קלט 102X102X3, אם אנו בוחרים פילטר (1X1X3) 1X1, אזי רוחב וגובה הקלט ישמרו אך כמספר הפילטרים בשכבה כך עומק הפלט.

נניח ובשלב כלשהו במודל שלנו הגענו מטריצה בעלת עומק גדול מהוד נניח 300 וברצוננו להפחית את מספר הערוצים, אבל לשמור על הגובה וברוחב של ה feature map, אנו יכולים להשתמש בקונבולוציה 1X1 וכמספר הפילטרים בשכבה כך יהיה עומק הפלט. פעולה זאת נקראת 'Dimensionality reduction'.



שימושים:

1. הפחתת מימדים/augmentation.
2. צמצום הפרמטרים ברשת.
3. הוספת אי-ליניאריות לרשת.
4. יצירת רשת עמוקה יותר על ידי שכבות "(ResNet-50)."Bottle-neck

GoogLeNet, Inception–V1

הזוכה בתחרות ILSVRC2014, השתמשה ב 1X1 convolution על מנת להפחית את המימדים. השיטה היא ביצוע קונבולוציות 1X1 שהיא שחות כבדה לפני הקונבולוציות הכבדות יותר 5X5, 3X3 ובכל להפחית את מספר הפרמטרים הכולל של הרשת.

לדוגמה: נניח ששכבת קונבולוציה מקבלת קלט של 28X28X192 ומפעילה עליו 5X5X32 פילטרים (כלומר 32 פילטרים בגודל 5X5) כמה פרמטרים יש לנו?

$$5*5*32*192+ 32= 153,632$$

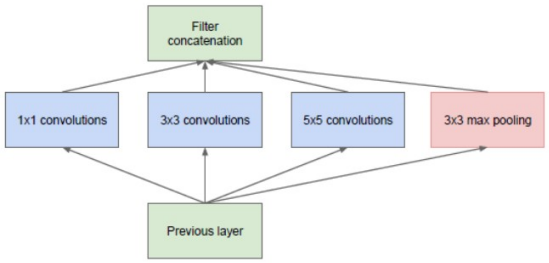
כעת לפני שכבת הפילטרים 5X5 נשים שכבת פילטרים 1X1X16 ולאחר מכן שכבת 5X5. כמה פרמטרים?

$$(1*1*192*16+ 16) + (5*5*32*16+ 32)= 15,920$$

על ידי הוספת שכבת קונבולוציה 1X1 לפני קונבולוציה 5X5 בעוד ששמרנו על האורך והגובה של ה feature map, מספר הפרמטרים קטן בפקטור של 10! זה יקטים את צרכי החישוב, ובסופו של דבר יהיה יעיל יותר.

Inception module:

Inception module נאיבי ללא 1X1convo:



(a) Inception module, naïve version

Inception Module (Without 1×1 Convolution)

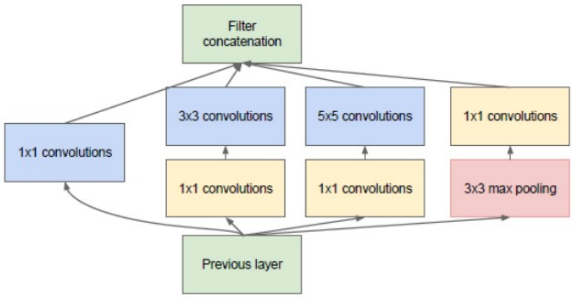
לפני כן, רשתות כמו AlexNet , VGG–16, בצעו קונבולוציות בגודל קבוע לכל שכבה.

הרעיון כעת הוא לאגד כמה features map בייחד על הקלט של השכבה הנוכחית:

1×1 conv, 3×3 conv, 5×5 conv, and 3×3 max pooling

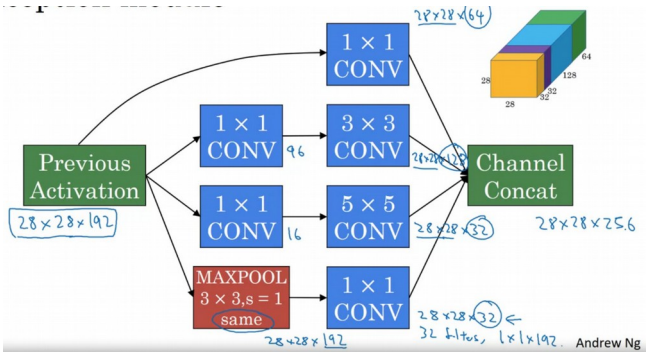
בדרך זו אנחנו מקבלים מספר features extractions עבור אותה שכבה. עם זאת, ללא קונבולוציית 1X1 כמו בתמונה שלמעלה מספר הפרמטרים בשכבה כזאת הוא עצום!

לפיכך, נכניס קונבולוציית 1X1 למודול על מנת להפחית את המימדים:

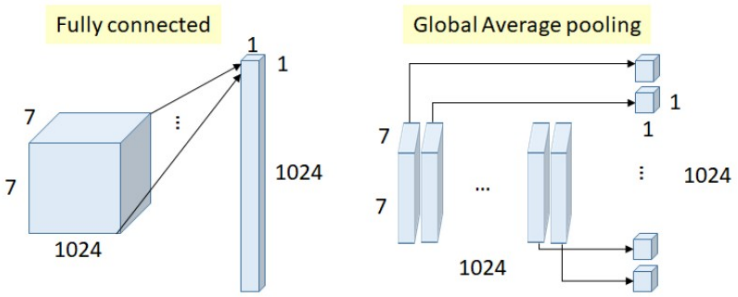


(b) Inception module with dimensionality reduction

Inception Module (With 1×1 Convolution)



Global average pooling:

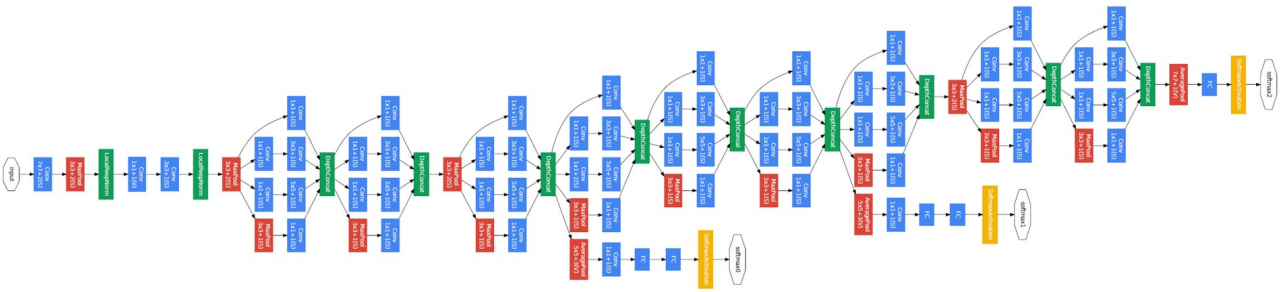


ברשתות הקודמות שראינו, בסוף הרשת משתמשים ב fully connected layers.
למשל בדוגמה למעלה מספר המשקולות הוא

$$7*7*1024*1024+1024=51,381,248$$

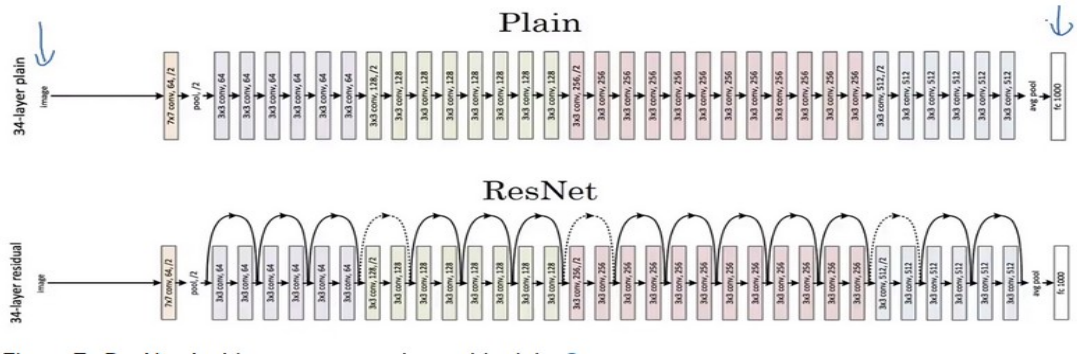
ב GoogLeNet נעשה שימוש ב Global average pooling לקראת סוף הרשת על ידי לקיחת ממוצע מכל feature map כמו שמתואר בתמונה. מספר המשקולות בשיטה זאת הוא 0. והכותב מצא כי המעבר מ FC ל Global average pooling משפר את הדיוק בכ-0.6%.

הארכיטקטורה הכללית:
לאחר הכרת יחידות הבסיס, נוכל לדבר על ארכיטקטורת הרשת הכוללת. הרשת כוללת 22 שכבות:



זה מודל עמוק יותר מקודמיו (אך לא כה עמוק בהשוואה ל ResNet שהגיע אחריו).

Residual Network (ResNets)



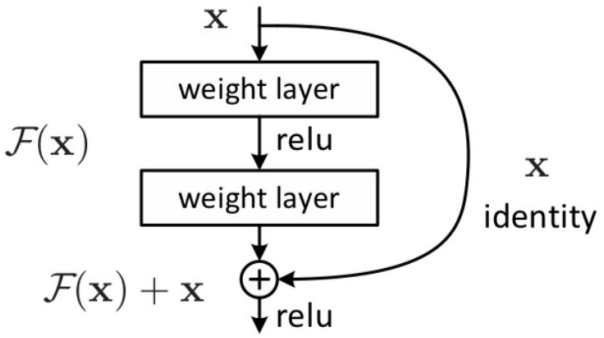
בשנת 2014 הומצאה ResNet וזכתה בתחרות ILSVRC2015. לאחר פריצת הדדרך הגדולה של הרשת AlexNet בתחרות הסיווג ILSVRC2012 ResNet השיג פריצת דרך נוספת. ResNet מאפשר לאמן מאות ואף אלפי שכבות נסתרות ועדיין משיג ביצועים משכנעים. באמצעות מודל זה שופרו רבים

מהיישומים של ראייה ממוחשבת, כמו זיהוי אובייקטים וזיהוי פנים. בשנים האחרונות, הרשותות הלכו ונעשו עמוקות יותר. אחד היתרונות העיקריים של רשת עמוקה מאוד הוא שהיא יכולה לייצג פונקציהה מורכבת מאוד. הבעיה היא שברשתות כה עמוקות היא עלול להיווצר מצב של vanishing gradient. במהלך תהליך ה back-prop, אנו מחשבים את הגרדיאנט של הרשת שלנו. כלומר, כיצד כל משתנה משפיע על ה ERROR, שאותו כאמור אנו מנסים למזער. עלינו לעדכן את המשקולות על מנת למזער את השגיאה. הבעיה ברשתות עמקות היא שלא רק הנוירונים שנמצאים ממש מתחת לשכבת הפלט הם אלה שתרמו אלא גם כל הנוירונים הנמצאים בתחילת הרשת. לכן עלינו לשנות את המשקולות הנמצאות התחילת הרשת. הבעיה היא שהמשקל המשמש לחיבור השכבות הנסתרות בעצמו נוסף לחישוב הרמה הבאה. וכאן מתעוררת הבעיה: כאשר מכפילים משהו במספר קטן שוב ושוב הערך שלנו יורד מהר מאוד.

הרשתות הרגילות כמו VGG-16 נקראות "plain networks" רשתות שטוחות. בתיאוריה, אנו מצפים כי ככל שהרשת תהיה עמוקה יותר, ככה היא תניב תוצאות טובות יותר. אך במציאות רשתות גדולות לרשתות גדולות מידי יש שגיאה גדולה יותר.

הפתרון: Residual Block / Identity block.

הרעיון הוא להוסיף shortcut או skip connection המאפשר זרימת מידע ביתר קלות משכבה אחת לשכבה הבאה. כלומר, מעקף של שכבת קונבולוציה והכנסת הפלט הקודת לפלט הבא. איך זה נראה?



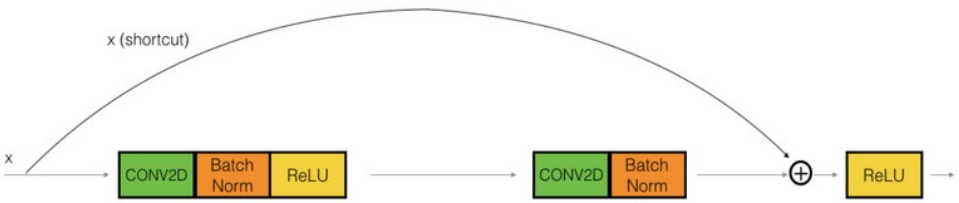
הערה:

הוספת השכבות הנוספות בצורה הזאת לא תפגע בביצועי המודל. מכיוון שהרשת תלמד בקלות פשוט להעתיק את בהפלט המקורי לפני הוספת ה Residual Block.

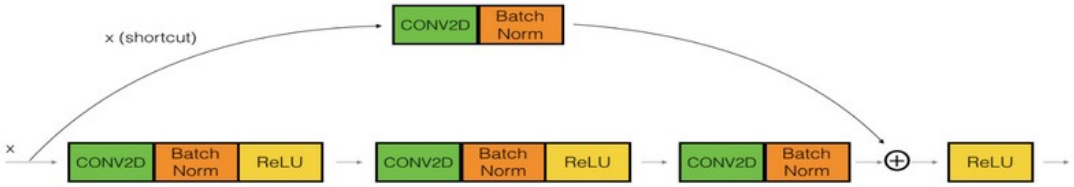
לפיכך, בזכות Residual Block / Identity block מובטח לנו שעל ידי הוספת שכבות נוספות ביצועי המודל לא יפחתו, והם עשויים לעלות. על ידי ערימת ResNet blocks אלו אחד על גבי השני, נוכל ליצור רשת עמוקה מאוד.

ישנם שני סוגים עיקריים של בלוקים ב- ResNet, תלוי בעיקר אם ממדי הכניסה/ פלט זהים או שונים.

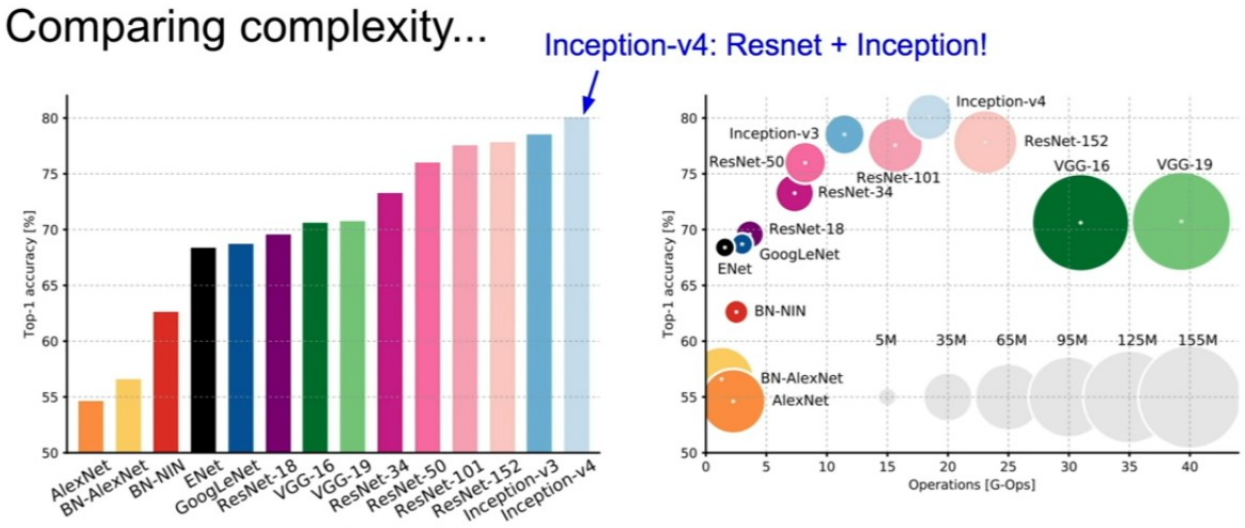
1. The identity block- בלוק הזהות הוא הדרך הסטנדרטית בשימוש ב ResNet והיא מיודעת למקרה בו ל input activation מימד זהה ל output activation.



2. The Convolutional block – אנו יכולים להשתמש בבלוק זב כאשר ממדי הקלט והפלט אינם תואמים. ההבדל הוא הוספת שכבת קונבולוציה בנתיב הקיצור.



השוואות



An Analysis of Deep Neural Network Models for Practical Applications, 2017.

Inception-V4: הכי מדוייק.
VGG: צורך הכי הרבה זיכרון/הרבה פעולות.
GoogLeNet: הכי יעיל.
AlexNet:מעט פעולות, הרבה זיכרון, דיוק נמוך.
ResNet-18: מבין המודלים שנסקרו הוא הכי מדוייק, צריכת זיכרון בינונית.