**יונתן בן אברהם, איתי חי,**

**שניר מרדכי, קארין וסנשטיין**

|  |
| --- |
|  |
| **Final Project- Machine Learning** |
|  |
|  |

**מבוא**

**פירוט המשימה:**

כחלק מאתגר עיבוד תמונה של גוגל (Google Landmark Retrieval Challenge), התבקשנו לבנות רשת נוירונים אשר יודעת לסווג תמונת נוף בין 100 קטגוריות שונות של נופים בעולם.

**המשימה התחלקה לשתיים:**

1. ליצור רשתות נוירונים אשר ידעו לסווג את התמונות ל-class-ים השונים.
   1. 3 רשתות Transfer Learning:
      1. Vgg16
      2. InceptionV3
      3. Resnet50
   2. רשת אחת מתוך כלל הרשתות הינה רשת אשר בנינו בעצמנו (Deep Neural Network)
2. ליצור מסווג KNN שלפי ה-Feature-ים של הרשתות שייצרנו ימצא את 10 התמונות הקרובות לו ביותר ויציג אותן.

**מוטיב לבחירת הרשתות :**

הרשתות שבחרנו להשתמש בהן אומנו זה מכבר על מאגר הנתונים של ImageNet. כלומר רשתות אלו יודעות לזהות אובייקטים בצורה גבוהה. לכן, נוסיף להן מספר שכבות שידעו להשתנות בהתאם לשינוי באתגר האופטימיזציה (אנחנו מסווגים landmarks במקום עצמים וחפצים). באופן זה עם ביצוע התאמות וניסויים נדרשים, נבדוק מה יהיה השילוב האופטימלי.

**כתובת הפרויקט ב-GitHub:** [**https://github.com/itaihay/Colman\_ML\_Google\_Landmark**](https://github.com/itaihay/Colman_ML_Google_Landmark)

**רשת מספר 1- Vgg16**

**פירוט הניסוי:**

בניסוי זה ננסה לקחת את רשת VGG16 הקיימת ולאמן אותה מחדש לטובת זיהוי אתרים.

רשת זו, בדומה לשאר הרשתות בפרוייקט שלנו, אומנו במקור על ImageNet. הרשת יודעת כבר לזהות עצמים דוממים יום-יומיים, והמטרה שלנו היא פשוט לגרום לה לזהות אתרים יותר טוב.

אתגר זה הוא הפשוט ביותר מבין שאר הניסויים מאחר ומדובר ברשת שנלמדה בשיעור. המטרה היא להראות את השינויים ב-Hyper Parameters ואת ההשפעה שלהם על אימון הרשת.

**שלבי ההרצה של ניסוי בודד:**

1. טעינת התמונות של ה- TrainingData וטעינת התמונות של ה- ValidationData
2. יצירת המודל (VGG16 עם שכבות Dense)
3. הגדרת הפרמטרים, פונקציית ה-Loss וה-optimizer.
4. יצירת ה callbacks עבור שמירת ההיסטוריה של תוצאות ה loss וה accuracy בסיום כל epoch + שמירת המשקלים.
5. אימון המודל.
6. יצירת הגרפים ומדידת תוצאות האימון.

**חלק ראשון- בניית רשת שמתבססת על VGG16:**

בניסוי מסוג Transfer Learning אנו מקפיאים את השכבות הקיימות ברשת, מאחר והן מאומנות כבר במקור.

לאחר אותן שכבות נוסיף כמה שכבות משלנו שאותם נאמן כדי לתת classification לתמונות לפי 100 ה-classים שלנו:

global\_average\_pooling2d (Gl (None, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense (Dense) (None, 1024) 525312

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 1024) 1049600

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_2 (Dense) (None, 512) 524800

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_3 (Dense) (None, 100) 51300

===============================================================

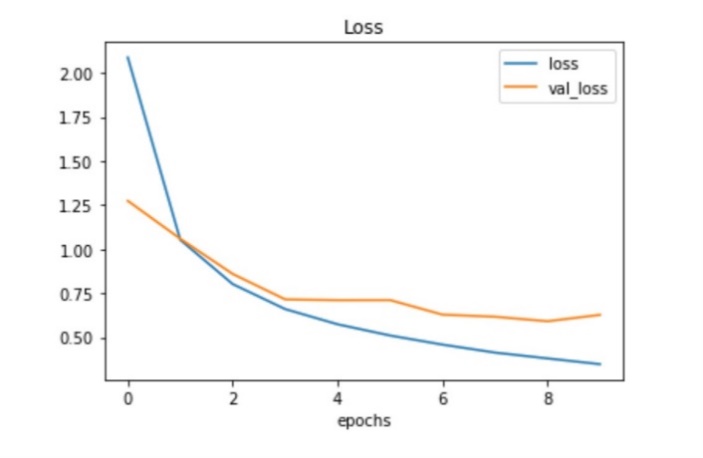
* Global Average Pooling  
  מטרת השכבה היא לנסות למנוע over fitting. היא מבצעת נרמול וממוצע לכל פילטר כך שמצורה של 4x4x512 נעבור לצורה של 1x1x512.
* שכבות ה-dense  
  שכבות fully connected שמטרתן לסווג יותר טוב את התמונה. אלו השכבות שנאמן כדי לקחת את היכולות של VGG16 בסיווג עצמים, ולהתאים אותה לסיווג אתרים.  
  גודל ה-output של שכבת ה-dense האחרונה היא לפי כמות ה-classים שלנו.

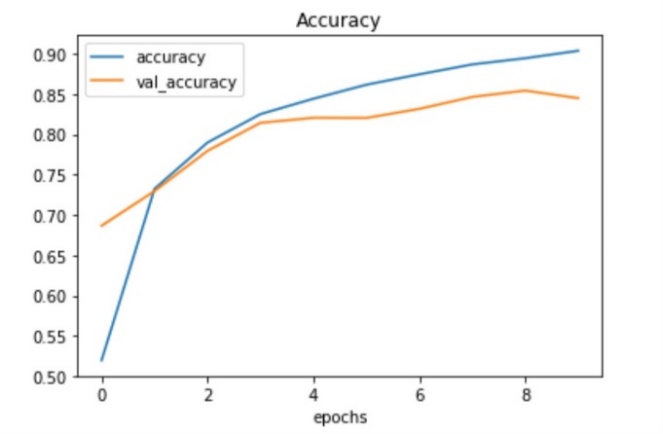
**חלק שני – ריצת ביקורת:**

בפעם הראשונה נבצע הרצה עם פרמטרים מסויימים שיהוו בסיס וביקורת לשאר הרצות הבדיקה:

**תוצאות הניסוי:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 0.9 | 10 | 0.0001 | SGD - Momentum | categorical\_crossentropy |





**מסקנות:**

ההרצה הראשונה הראתה תוצאות טובות.

loss: 0.3470   
acc: 0.9036   
val\_loss: 0.6268   
val\_acc: 0.844

כעת נריץ עוד הרצות כשבכל אחת נשנה פרמטר אחר.

**חלק שלישי – שינוי ה-LR:**

בניסוי הזה נרצה לראות את ההשפעה שיש ל-LR על ריצת הלמידה. ביצענו שתי ריצות: אחת עם LR מאוד גבוה, והשניה עם LR מאוד נמוך.

ניסוי ראשון

בניסוי הראשון הגדרנו LR = 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 0.9 | 10 | 1 | SGD - Momentum | categorical\_crossentropy |

בתאוריה כמובן שלא מדובר בהגדרה סבירה שכן לא יהיה ניתן לשפר בכלל את המשקולות, והתוצאות בהתאם:

loss: 15.8087   
acc: 0.0192  
val\_loss: 15.8096   
val\_acc: 0.01

|  |  |
| --- | --- |
|  |  |

בניסוי השני:

בניסוי השני הגדרנו LR = 0.00001.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 0.9 | 10 | 0.00001 | SGD - Momentum | categorical\_crossentropy |

loss: 0.8664  
acc: 0.7765  
val\_loss: 0.9933  
val\_acc: 0.7424

|  |  |
| --- | --- |
|  |  |

**מסקנות:**בהתאם להרצת הביקורת, ניכר כי ערך קצב הלמידה צריך להיות נמוך, אך לא נמוך מידי. הורדה קיצונית שלו מוריד את תוצאות הניסוי, אבל כן מקטין את ה-overfitting (ערך הval קרוב לערך הלמידה).

**חלק רביעי – שינוי ה-momentum:**

בחלק זה ננסה לשנות את פרמטר הממונטום לשני הכיוונים.   
הפרמטר גורם לכך ששינוי של המשקולות יהיה בצורה פחות קיצונית לכיוון נגדי.

נראה בשתי התוצאות את ההשפעה של הערך על התוצאות הסופיות.

ניסוי 1:

בניסוי הראשון הגדרנו momentum = 1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 1 | 10 | 0.0001 | SGD - Momentum | categorical\_crossentropy |

loss: 15.4690  
acc: 0.0403  
val\_loss: 15.4740  
val\_acc: 0.0400

|  |  |
| --- | --- |
|  |  |

ניסוי 2:

בניסוי השני הגדרנו momentum = 0.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 0 | 10 | 0.0001 | SGD - Momentum | categorical\_crossentropy |

loss: 0.8820  
acc: 0.7730  
val\_loss: 0.9988  
val\_acc: 0.7529

|  |  |
| --- | --- |
|  |  |

**מסקנות**

ניתן לראות בעזרת הניסוי ובעזרת ניסוי הביקורת כי יש השפעה חיובית לפרמטר המומנטום, אך הוא לא נדרש כדי להגיע לתוצאות סבירות.

בהתאם לניסוי הביקורת, הלמידה הייתה יותר איטית כשלא נעזרנו במומנטום, אך מנגד כשהגדרנו מומנטום שאיננו קטן מ-1 אז לא היה ניתן לבצע אימון.

**חלק חמישי – שינוי ה-batch size:**

בחלק זה נשנה את ה-batch size לשני כיוונים.

שינוי ה-batch size גורם לשינוי בקצב צעדי הלמידה בכך שמתבצע ממוצע ליותר תמונות בכל קפיצה (ככל שה-batch size גדל).

ניסוי 1:

בניסוי הראשון הגדרנו batch size = 32.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 0.9 | 32 | 0.0001 | SGD - Momentum | categorical\_crossentropy |

loss: 0.5332  
acc: 0.8560  
val\_loss: 0.7049  
val\_acc: 0.8224

|  |  |
| --- | --- |
|  |  |

ניסוי 2:

בניסוי השני הגדרנו batch size = 2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 0.9 | 2 | 0.0001 | SGD - Momentum | categorical\_crossentropy |

loss: 0.3029  
acc: 0.9150  
val\_loss: 0.5161  
val\_acc: 0.8712

|  |  |
| --- | --- |
|  |  |

**מסקנות**

ניתן לראות מהתוצאות כי אומנם ה-accuracy עלה, אך גם עלה ה-overfitting כשהקטנו את ה-batch size.

ניתן ללמוד שנדרש להשתמש בערך ביניים ולשלב אותו עם עוד ערכים כדי לקבל את המיטב משני העולמות.

**חלק שישי – הרצה מלאה:**

בשלב האחרון ניקח את מסקנות ההרצות הקודמות וננסה להגיע לתוצאה טובה.

המטרה היא לנסות להשתמש בפרמטרים שונים מהרצת הבסיס ולראות אם התוצאות משתפרות.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Decay | Learning Rate | Optimizer | Loss Function |
| 20 | 0.9 | 32 | 0.0002 | 0.0001 | SGD - Momentum | categorical\_crossentropy |

loss: 0.5013  
acc: 0.8640  
val\_loss: 0.7109  
val\_acc: 0.8219

|  |  |
| --- | --- |
|  |  |

**מסקנות**

בהרצה הנ"ל נראה כי יש אחידות טובה בין קבוצת הבדיקה לקבוצת הלמידה ואין overfitting גדול.

מצד שני נראה כי התוצאות פחות גבוהות מהתוצאות שקיבלנו בחלק מההרצות.

ניתן להניח כי נדרשות הרצות נוספות כדי לבצע fine tuning למודל, אך שבכללי הראינו שניתן לקחת את VGG16 וללמד אותו להבדיל בין עצמים שאינם מ-ImageNet.

**רשת מספר 2- Resnet 50**

**פירוט הניסוי:**

בעיה זו של הוצאת Feature-ים מתמונה וסיווג התמונה ל-class המתאים, היא בעיה דומה מאוד לאתגר ש-imageNet מציג, אשר מסווגת בעיקר בעלי חיים, צמחים ועצמים דוממים (לא נופים כמו באתגר שלנו).

מהסיבה הזו החלטנו שדרך טובה לפתור את האתגר שלנו תהיה ביצוע Transfer Learning מרשת שכבר אומנה באתגר של imageNet. הרשת שבחרנו לצאת ממנה היא רשת ResNet50.

**שלבי ההרצה של ניסוי בודד:**

1. טעינת התמונות של ה- TrainingData וטעינת התמונות של ה- ValidationData
2. טעינת המודל ResNet50
3. הוספת שכבות ה-Dense האחרונות בכדי ליצור את הפלט הרצוי (מערך של 100 מחלקות עם ציונים עבור כל אחד מהם לתמונה שהתקבלה)
4. הגדרת פונקציית ה loss, ה optimizer וה Hyper Parameters הרלוונטים לאותו optimizer (momentum, learning rate וכו')
5. יצירת ה callbacks עבור שמירת ההיסטוריה של תוצאות ה loss וה accuracy בסיום כל epoch + שמירת המשקלים.
6. אימון המודל.
7. יצירת הגרפים ומדידת תוצאות האימון.

**חלק ראשון- קיבוע ה-Learning Rate:**

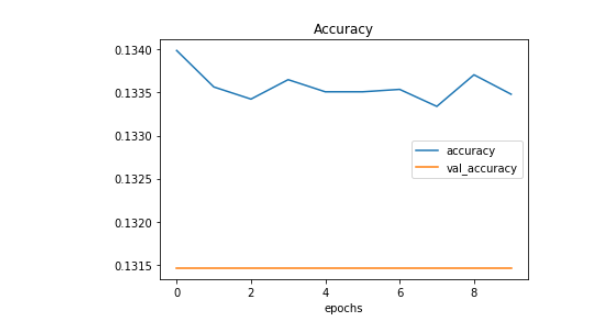
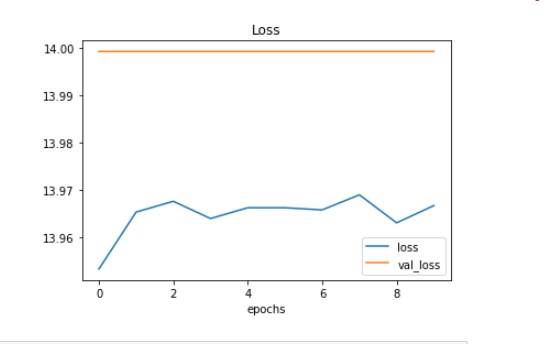
בשלב זה ננסה למצוא Learning rate מתאים עבור ה optimizer שבחרנו (Stochastic Gradient Descent). על פי מה שנלמד בשיעור, Learning rate יהיה בד"כ בטווח הערכים בין 0.00001 ל- 1.

ניקח את הקצוות תחילה, ולאחר מכן ערך אמצעי ונראה את תוצאות האימון עבור כל אחד מהערכים.

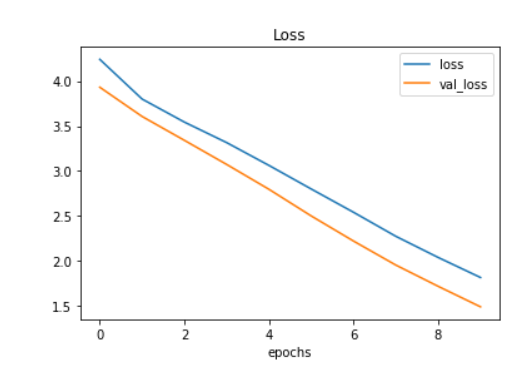
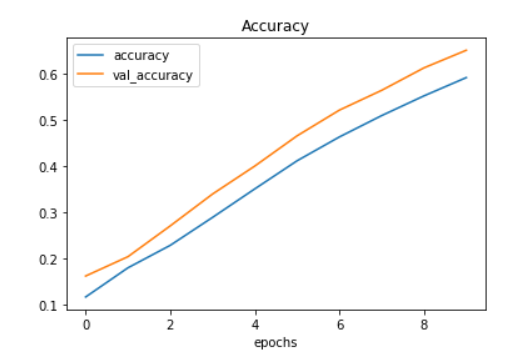
בנוסף ניקח כמות קטנה יחסית של epochs מכיוון שבשלב זה נרצה בעיקר לראות מגמות של איך אנחנו מתפקדים עם learning rates שונים.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 0.9 | 10 | 1 | SGD - Momentum | categorical\_crossentropy |

**סיכום תוצאות:**

****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 0.9 | 10 | 0.00001 | SGD - Momentum | categorical\_crossentropy |

****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 0.9 | 10 | 0.001 | SGD - Momentum | categorical\_crossentropy |

****

**מסקנות:**

* ניתן לראות שכאשר הstep היה גדול מידי (Learning Rate = 1), לאחר כל batch שהסתיים הרשת עשתה שינוי גדול מידי על המשקולות. השינוי הזה גרם לכך שנפספס את נקודות המינימום שאנו מחפשים על ה loss. ה-learning Rate הגבוה גרם לכך שהצעד שאנחנו עושים בכיוון המנוגד ל- Gradient יהיה גדול מידי, מה שגרם לשינוי לא נכון של המשקולות על הרשת.
* ניתן לראות שכאשר ה step היה קטן מידי (Learning Rate = 0.00001) האימון על הרשת גם לא הצליח. אך הגרף נראה שונה לגמרי, הוא כן מתקדם לכיוון הנכון פשוט בצורה איטית מאוד. לכן 10 epochs בלבד לא מספיקים לו.
* כאשר ערך ה Learning Rate היה 0.01 ניתן לראות שתוצאות האימון לאחר 10 epochs היה לא רע בכלל. הגענו לכמעט 90% Accuracy על קבוצת הוולידציה שלנו.

**חלק שני- קיבוע ה-Batch Size וה-Momentum:**

כעת נרצה לקבע את שני ה-Hyper Parameters הנוספים: ה-Batch Size וה-Momentum.

* לאחר מחקר באינטרנט מצאנו שאת המומנטום יש להגדיר בין 0 ל 1 (על פי הנוסחה המתמטית). בנוסף ראינו שלרוב מגדירים את המומנטום בערך 0.9. בניסויים שלנו ננסה את ערך זה ובנוסף ערך קטן יחסית של 0.5.
* ה-Batch Size הוא משתנה יותר עדין ותלוי גם בגודל ה-Training Data שלנו. לרוב הוא גם יהיה תחום בין הערך 1 וכמה מאות בודדים. ערכים נמוכים ייגרמו לכך שאימון הרשת ידרוש פחות Epochs, מכיוון שבכל epoch נעשה כמות גדולה יותר של שינויים על המשקולות (שינוי נעשה בסיום כל batch). אבל, Batch קטן עלול לעשות יותר תנועות בכיוונים לא נכונים, וזאת מכיוון שהשינוי שייעשה יהיה על מעט מידי דוגמאות ולא יהיה מייצג. לעומת זאת batch גדול מידי יאריך משמעותית את זמן האימון מכיוון שנקבל מעט מאוד שינויים על המשקולות בכל epoch דבר שידרוש כמות גדולה יותר של epochs מה שיגרור זמן אימון ארוך הרבה יותר. נבדוק את האימון על Batch Size של 10, 16, 32. (ממגבלות של הזיכרון שיש לנו על ה GPU לא נוכל להריץ Batch גדול מ 32)

**סיכום תוצאות:**

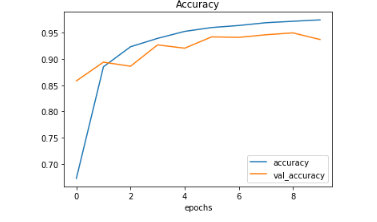
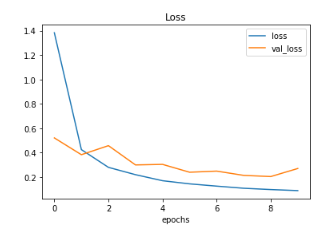
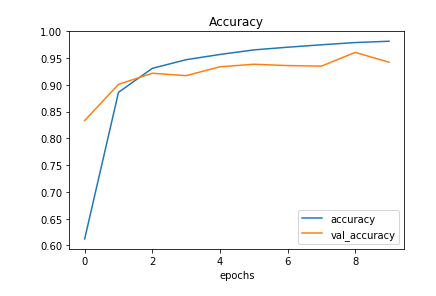
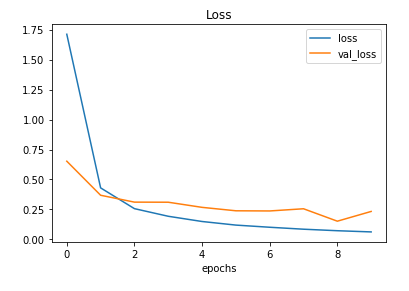
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 0.9 | 10 | 0.001 | SGD - Momentum | categorical\_crossentropy |



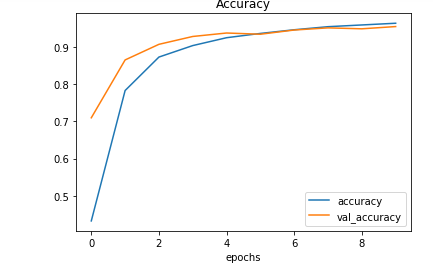
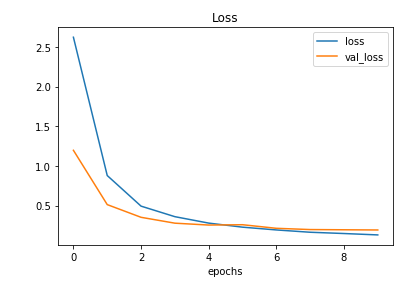


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 0.9 | 16 | 0.001 | SGD - Momentum | categorical\_crossentropy |

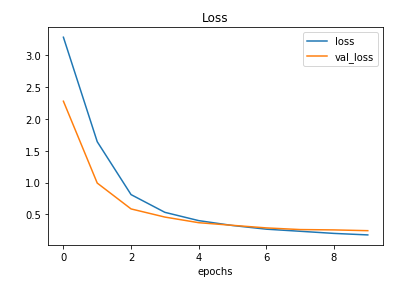
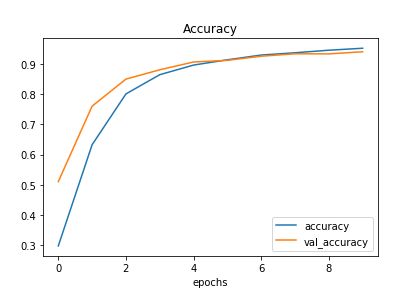
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 0.9 | 32 | 0.001 | SGD - Momentum | categorical\_crossentropy |

****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 0.5 | 16 | 0.001 | SGD - Momentum | categorical\_crossentropy |



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 0.5 | 32 | 0.001 | SGD - Momentum | categorical\_crossentropy |



**מסקנות:**

אפשר לראות מגמה מאוד מעניינת בהקשר של המומנטום- בניסויים בהם המומנטום היה 0.5 המרווח בין ה Training Data ל Validation Data היה קטן מאוד וכמעט אפסי. לא הצלחנו למצוא צידוקים לזה מהחומרים שקראנו אבל זה לא נראה כמו צירוף מקרים. להערכתנו יכול להיות שזה קשור לעובדה שעבור מומנטום קטן אנחנו עושים "צעדים" קטנים יותר בכיוון ה Gradient.

בנוסף, ניתן לראות שעבור המומנטום הגבוה הגרף נע בצורה יותר חדה ומגיע כבר ב Epochs הראשונים לאחוזי דיוק יחסית גבוהים ולאחר מכן קצב השיפור קטן בצורה חדה.

לכן, יכול להיות שעבור המומנטום (כמו ה-Learning Rate) נרצה להתחיל את האימון בערך גבוה יחסית של המומנטום ולאחר מכן כשנרצה "לגרד" את אחוזי הדיוק הגבוהים נקטין את המומנטום.

עבור גודל ה Batch אפשר לראות מגמה ברורה ולא מפתיעה, שכאשר גודל ה Batch היה מאוד גבוה אימון הרשת היה יותר איטי אבל באופן מובהק עם פחות טעויות. אפשר לראות את זה בצורה החלקה יחסית שיש לגרפים כאשר גודל ה Batch הוא 32. כאשר גודל ה batch הוא 10 ו 16 קל לראות שהיו epochs שבסוף ערך ה-accuracy אפילו ירד.

כמו כן ניתן לראות שעבור Batch בגודל 10 קיבלנו ביצועים פחות טובים בצורה משמעותית. לכן נשתמש בעיקר בגדלי ה batch של 16 ו -32 בהמשך האימון.

כצפוי, ככל שגודל ה batch היה גדול יותר לקח יותר זמן לאמן את המודל, מכיוון שבכל Epoch עשינו פחות שינויים על המשקולות. מצד שני השינויים היו יותר מדויקים מכיוון שעשינו ממוצע של השינוי הנדרש על יותר דגימות.

**חלק שלישי – אימון הרשת:**

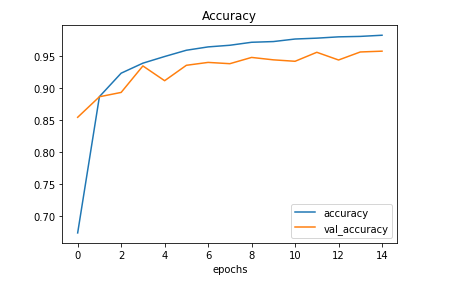
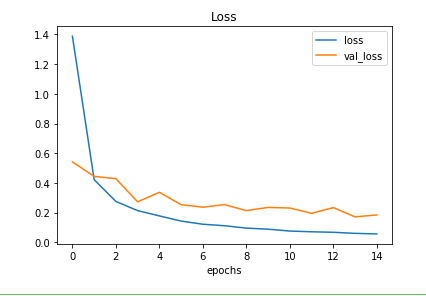
כעת נאמן את הרשת עם ערכי ה Hyper Parameters שמצאנו כמתאימים ביותר לאימון הרשת:

* Learning Rate = 0.001
* Momentum = 0.9
* Batch Size = 16

את האימון הזה של הרשת נבצע עם כמות גדולה יותר של Epochs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 0.5 | 16 | 0.001 | SGD - Momentum | categorical\_crossentropy |

**סיכום תוצאות:**



**מסקנות:**

ניתן לראות שהגענו לאימון לא רע בכלל של כמעט 95% דיוק על קבוצת ה validation.

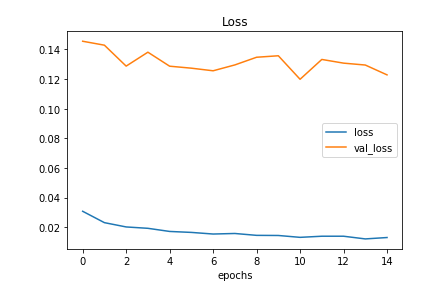
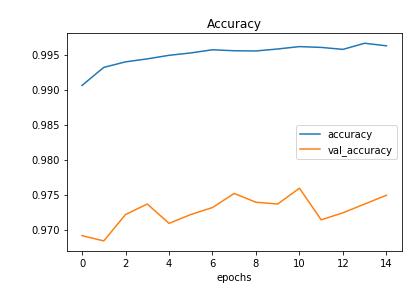
בשלב הבא נקטין את ה-Learning Rate וה-Momentum ונגדיל את ערך ה-BatchSize מכיוון שבאחוזים הללו יש סיכוי יותר גבוה להתקדם בכיוון הלא נכון

**חלק רביעי – הקטנת ה Learning Rate והמומנטום והגדלת ה Batch Size:**

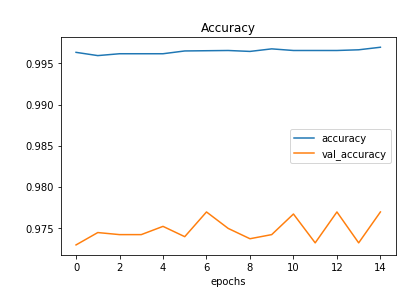
כעת הרשת שלנו נמצאת עם למעלה מ 90% accuracy על ה validation שלנו. מה שיגרום לכך שהסיכוי שנעשה צעד בכיוון הלא נכון גבוה יותר. בשלב זה כדי לאמן את הרשת בצורה טובה יותר, נבצע עוד 3 הרצות בהן נקטין בצורה משמעותית את ה Learning Rate שלנו, נגדיל את ה Batch Size ונוריד את המומנטום.

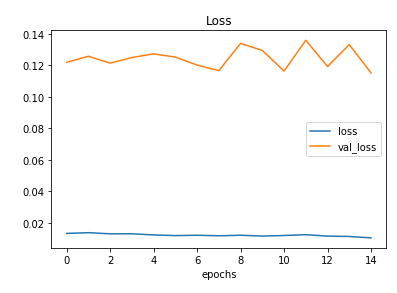
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 15 | 0.5 | 24 | 0.001 | SGD - Momentum | categorical\_crossentropy |

**סיכום תוצאות:**

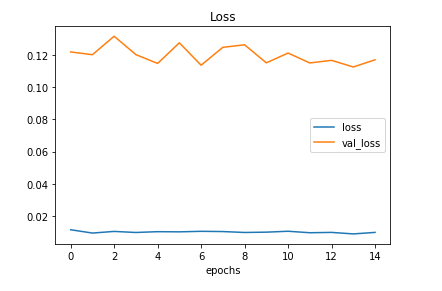
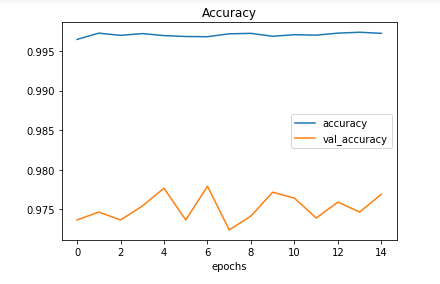


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 15 | 0.5 | 24 | 0.006 | SGD - Momentum | categorical\_crossentropy |





|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 15 | 0.5 | 32 | 0.0001 | SGD - Momentum | categorical\_crossentropy |

****

**מסקנות:**

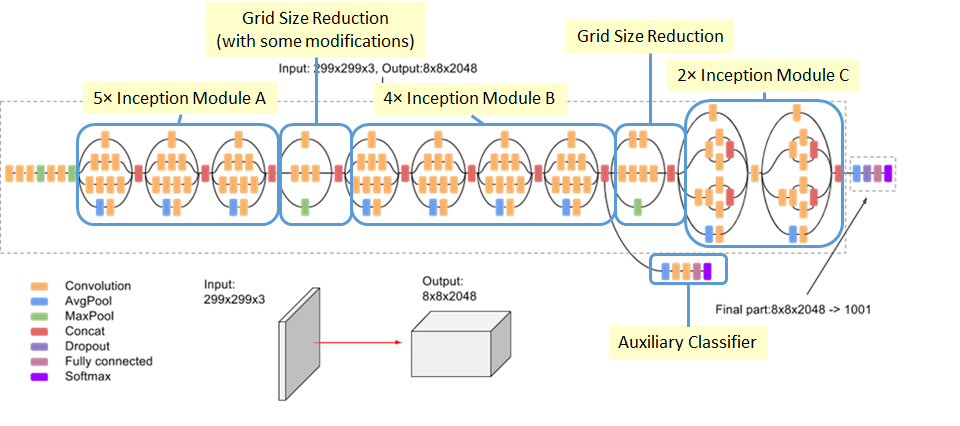
הגענו לאחר הניסויים לאחוז דיוק מצוין של 97.5% דיוק על ה Validation Data.

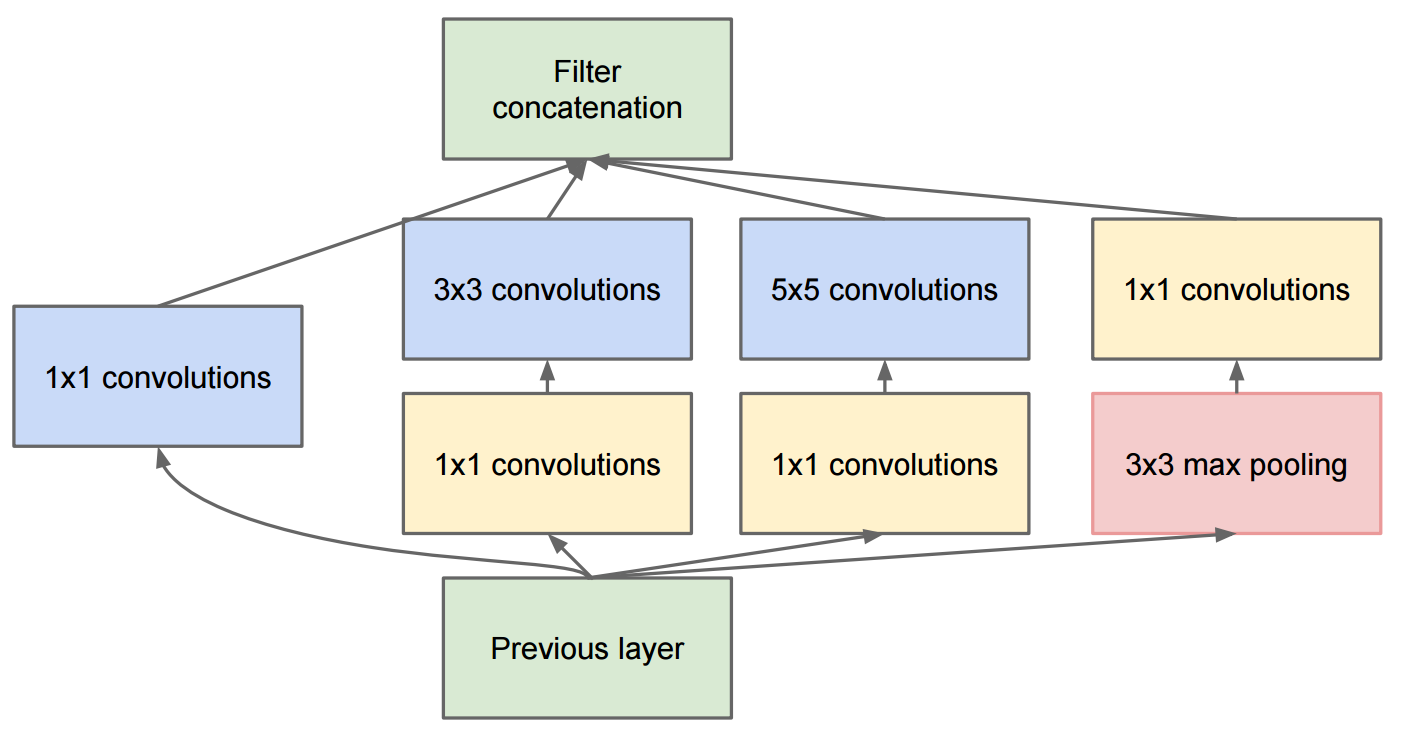
**רשת מספר 3- InceptionV3**

**פירוט הניסוי:**

ברשת הקודמת שמנו דגש על ה-Hyper parameters שנבחרו לאימון כשהנחת הייסוד שלנו הינה שהרשתות אתן נבחר לעבוד מאומנות היטב, ובחירה נכונה של פרמטרים תוביל לתוצאות טובות במידה ובעיית האופטימיזציה דומה. בניסוי זה בחרנו להשתמש ברשת InceptionV3.

InceptionV3, הינה גרסה מאוחרת יותר של GoogleNet. היתרון ברשת זו היא שהיא דורשת הרבה פחות פרמטרים לחישוב לעומת רשתות מוכרות אחרות, אך למרות זאת מצליחה להשתוות אליהן ברמת המורכבות אליה היא מצליחה להגיע. הרשת מבצעת זאת ע"י פירוק לגורמים של חישובי הקונבולוציה ובכך מקטינה את מספר הפרמטרים המחושבים.

בחרנו להשתמש ברשת זו, משום שזכתה במקום הראשון בתחרות ILSVRC 2015 בקטגורית Image Classification, התואמת לבעיית האופטימיזציה אתה אנו מתמודדים. כמו כן, עקב שיקולי ביצועים, כמות הפרמטרים הנדרשים לחישוב קטנה יותר, לכן יהיה קל יותר לבצע אימונים על הרשת.



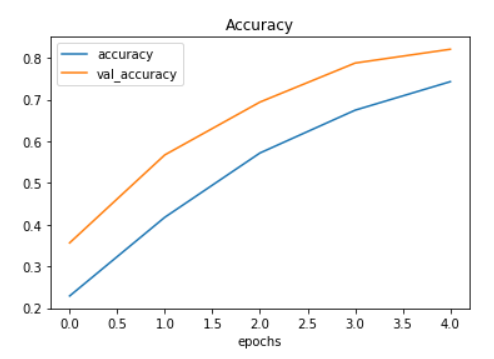
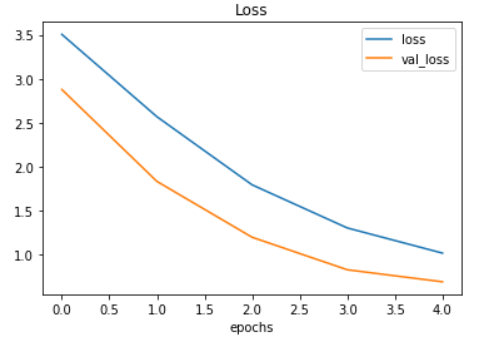
**שלבי ההרצה של ניסוי בודד:**

1. טעינת התמונות של ה- TrainingData וטעינת התמונות של ה- ValidationData
2. טעינת המודל InceptionV3
3. הוספת שכבות ה-Dense האחרונות בכדי ליצור את הפלט הרצוי (מערך של 100 מחלקות עם ציונים עבור כל אחד מהם לתמונה שהתקבלה)
4. שינוי והגדרת פרמטרים מתאימים בהתאם לניסוי / בדיקה.
5. יצירת ה callbacks עבור שמירת ההיסטוריה של תוצאות ה loss וה accuracy בסיום כל epoch + שמירת המשקלים.
6. אימון המודל.
7. יצירת הגרפים ומדידת תוצאות האימון

**חלק ראשון קיבוע ה Learning Rate:**

ראשית, כדי להתחיל מנקודת מוצא טובה יותר, ננסה למצוא את הLearning Rate המתאים ביותר בהתייחס לאלגוריתם InceptionV3. ניקח את SGD בשלב זה כ- Optimizer שלנו.

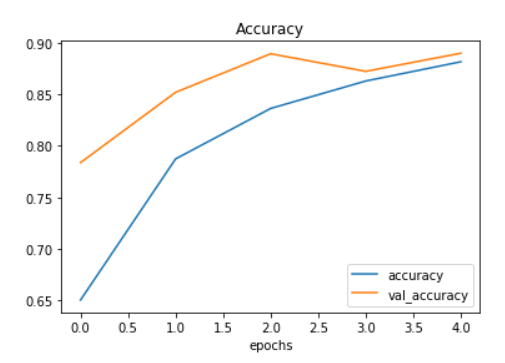
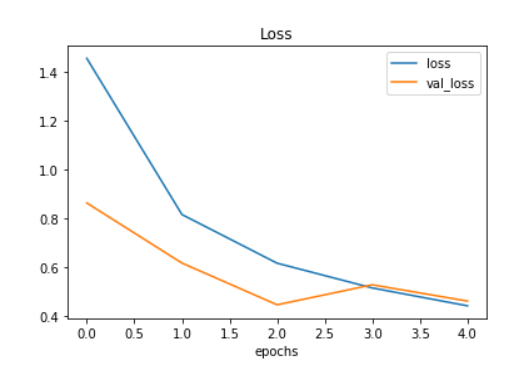
נשתמש בכמות קטנה יחסית של epochs מכיוון שבשלב זה נרצה בעיקר לראות מגמות על האופן בו האופטימיזציה מתבצעת עם ה- learning rates שונים.

**בדיקה 1-**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 5 | 0.9 | 10 | 0.0001 | SGD - Momentum | categorical\_crossentropy |

**בדיקה 2-**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 5 | 0.9 | 10 | 0.001 | SGD - Momentum | categorical\_crossentropy |



**סיכום תוצאות:**

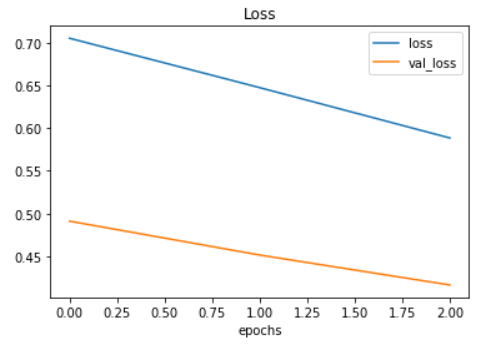
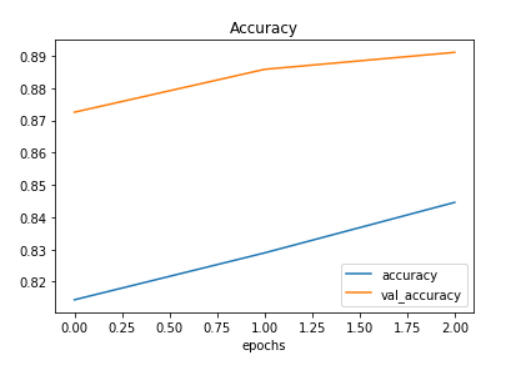
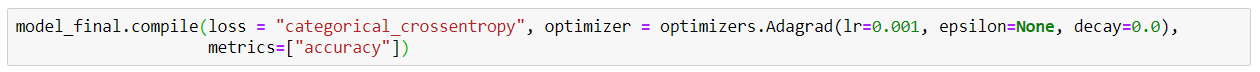
נראה כי ב-5 הרצות Learning Rate של 0.0001 קטן מידיי ולא מצליח להגיע במהירות לנקודת המינימום. נמשיך להתקדם עם 0.001.

**חלק שני – בחינת שיטות אופטימיזציה שונות**

**Adagrad optimizer**

כעת ננסה לאתגר מעט את אלגוריתם הלמידה ולהתאים אותו למאגר התמונות. ניתן לראות בברור שישנן תמונות באיכות נמוכה יותר מאחרות, לכן למרות שאנחנו מבצעים נורמליזציה ומגדירים את העיבוד על התמונות בגודל זהה, עדיין מתקבלות תמונות באיכויות שונות. לכן, במקום להשתמש ב Optimizer של SGD, נשתמש ב Adagrad. Adagrad ידוע ביכולת שלו להתמודד עם מידע מדולל, בכך שמבצע עדכונים עם שיעור למידה נמוך עבור פרמטרים הבאים לידיי ביטוי בפיצ'רים בתדירות גבוהה ושיעור למידה גבוהה עבור פיצ'רים בתדירות נמוכה. באופן זה אנו **משערים** כי האלגוריתם צריך לדעת להתמודד עם מידע דליל יותר, אשר יתכן ורלוונטי במקרה שלנו.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epochs | momentum | Learning Rate | Optimizer | Loss Function |
| 5 | 0.9 | 0.001 | Adagrad | categorical\_crossentropy |

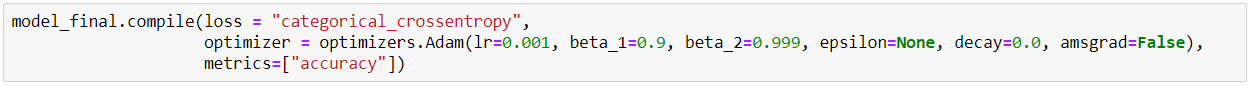


**סיכום תוצאות:**

לאחר 10 הרצות בלבד, הצלחנו להגיע לאחוז דיוק של 92% על קבוצת הבדיקה.

loss: 0.3668 - acc: 0.8985 - val\_loss: 0.3195 - **val\_acc: 0.9237**

**Adam Optimizer:**

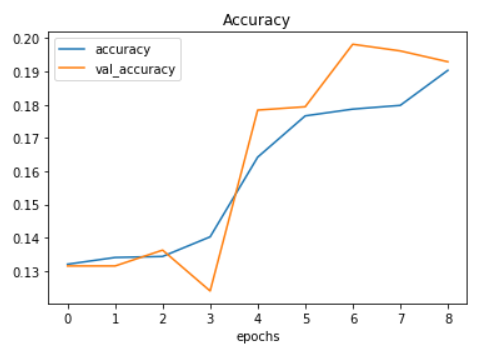
כעת, נרצה לבדוק Optimizer שונה.

עפ"י בדיקות נראה שAdam הוא Optimazer בעל ביצועים גבוהים.

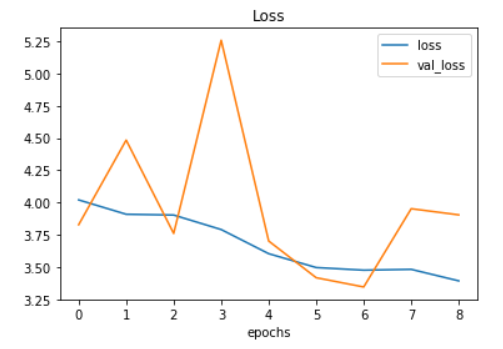
נאתחל אותו בערכי ברירת המחדל וננסה להריץ את אלגוריתם InceptionV3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epochs | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 10 | 0.001 | Adam | categorical\_crossentropy |

נתבונן בתוצאות האלגוריתם :



**סיכום תוצאות:**

תוצאות האלגוריתם אינן מספקות כלל ומראות חוסר יציבות גבוהה.

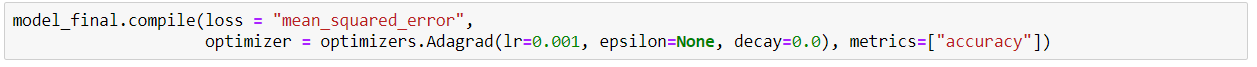
ככל הנראה הנושא נוצר עקב שילוב בעייתי של hyperparameters או ביצוע התאמות ברמת הרשת הנדרשות להרצה של Adam כ Optimizer של InceptionV3 שלא בוצעו כאן. נחזור "לחוף המבטחים" שלנו וננסה להגיע לחקור שינויים אחרים לביצוע.

**חלק שלישי – שינוי פונק' ה-Loss:**

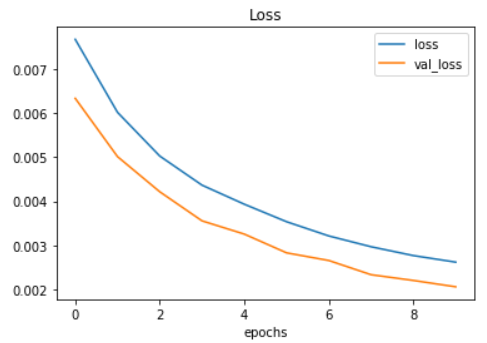
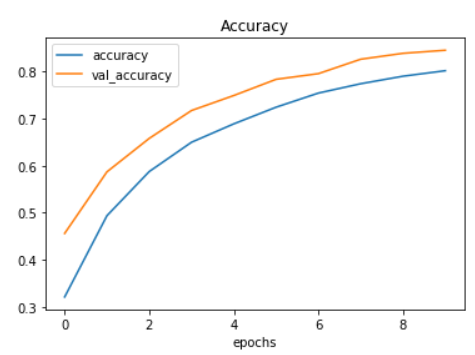
**Mean\_Squared\_Error Loss Function :**

כעת נתבונן על ההשפעה של פונקציות ה-Loss השונות על האלגוריתם שלנו.

אומנם mean\_squared\_error היא פונק' הפסד המשמשת לבעיות רגרסיה, אך היא מוכרת מאוד. מתוך סקרנות מחקרית, נבדוק מה יהיו הביצועים שלה על המודל שבנינו.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epochs | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 10 | 0.001 | Adagrad | mean\_squared\_error |



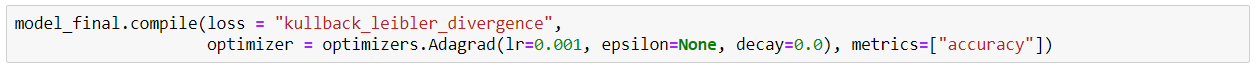
**סיכום תוצאות:**

לא הצלחנו להתעלות על התוצאות הקודמות שקיבלנו.

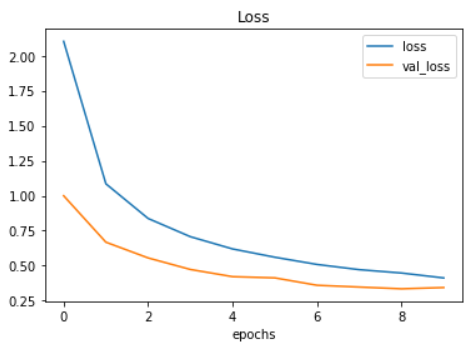
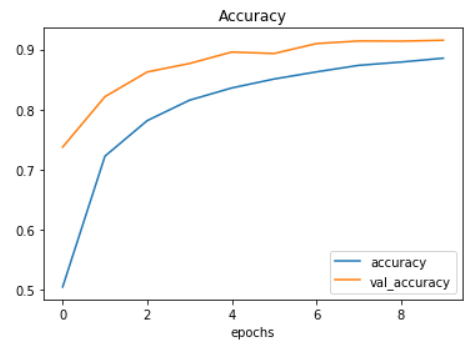
ננסה לחשוב על פונק' הפסד אחרת, שיתכן ותשפר את ביצועי המודל על ה- validation\_data.

**Kullback\_Leibler\_Divergence Loss Function:**

כעת ננסה להשתמש בפונק' הפסד שונה. פונק' זו דומה מאוד בהתנהגותה לפונק' ההפסד המוכרת cross-entropy, אך זו משמשת ללמידה וסיווג של בעיות מורכבות יותר מסיווג מרובה (multi-class classification). מעניין לראות כיצד תשפיע על פתרון בעיית האופטימיזציה בפניה אנו עומדים.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epochs | Batch Size | Learning Rate | Optimizer | Loss Function |
| 10 | 10 | 0.001 | Adagrad | kullback\_leibler\_divergence |



**סיכום תוצאות:**

הגענו לאחוז דיוק של 92%. ניתן לראות שהתוצאות טובות ואיכותיות אך לא הצלחנו להתעלות על ההרצה הטובה ביותר שביצענו.

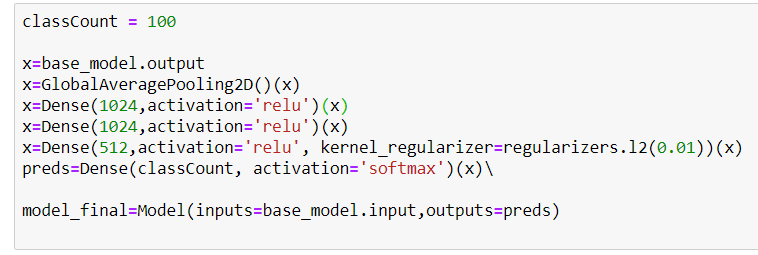
loss: 0.4124 - acc: 0.8854 - val\_loss: 0.3438 - val\_acc: 0.9152

**חלק רביעי – הוספת רגולריזציה**

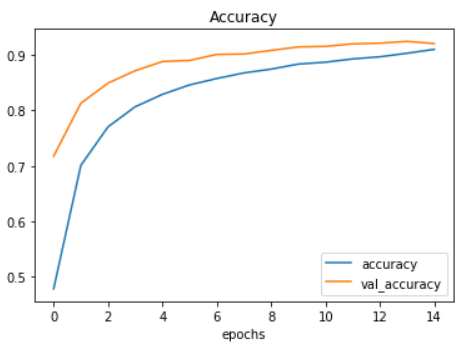
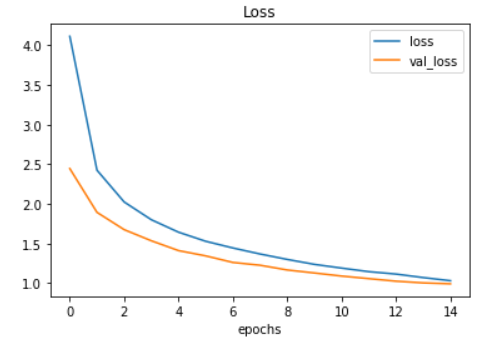
רגולריזציה היא טכניקה המאפשרת למנוע overfitting ע"י הענשה על סיבוכיות שנוצרת במודל, מתוך המחשבה כי אלו ימנעו מהמודל להיווצר כראוי, למרות שהוא מגיע לאחוזי הצלחה גבוהים על קבוצת האימון (training data).

אומנם, לא נראה כי נתקלנו עד כה במצב של overfitting מורגש או נראה לעין, אך נרצה לבדוק מה תהיה ההשפעה של שינוי זה על הרשת שלנו. לכן נוסיף באחת משכבות ה Dense - Regularizer, אשר ינסה להקטין את ה overfitting בתהליך הלמידה.

בנוסף לתוספת, נגדיל את מספר הEpochs שנריץ שכן נרצה לאמן מספק את הרשת משיקולי בדיקת התכנות לoverfitting בכל זאת.

להלן ההוספה שהתבצעה: 

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Epochs | Regularization param | Regularizer | Batch Size | Learning Rate | Optimizer | Loss Function |
| 15 | 0.01 | L2 | 10 | 0.001 | Adagrad | categorical\_crossentropy |

**סיכום תוצאות:**

הגענו לאחוז דיוק של 92.5%. עפ"י המשוער, ההשפעה לא הייתה משמעותית כל כך, אם בכלל הייתה.

למרות זאת הצלחנו להגיע לאחוז דיוק גבוהה מאוד המביא לידי ביטוי שילוב טוב ותוצאות טובות. loss: 1.0297 - acc: 0.9101 - val\_loss: 0.9908 - **val\_acc: 0.9244**.

**רשת מספר 4- Deep Neural Network**

**פירוט הניסוי:**

בניסוי זה ננסה ליצור רשת חדשה מאפס שתדע לטפל בבעיית זיהוי הנופים.

אתגר זה שונה מהניסויים בהם השתמשנו ב- Transfer Learning, מכיוון שכעת נצטרך לקבוע גם את עומק הרשת, את סוגי השכבות, הסדר שלהן וכדומה.

**שלבי ההרצה של ניסוי בודד:**

1. טעינת התמונות של ה- TrainingData וטעינת התמונות של ה- ValidationData
2. יצירת המודל (שכבות קונבולוציה + שכבות אקטיבציה relu, שכבות MaxPooling, שכבות Dense, שכבות Normalization וכו’).
3. הגדרת פונקציית ה loss, ה optimizer וה Hyper Parameters הרלוונטים לאותו optimizer (Batch Size, learning Rate)
4. יצירת ה callbacks עבור שמירת ההיסטוריה של תוצאות ה loss וה accuracy בסיום כל epoch + שמירת המשקלים.
5. אימון המודל.
6. יצירת הגרפים ומדידת תוצאות האימון.

**חלק ראשון- בניית הרשת הבסיסית:**

בשלב זה נבנה רשת בסיסית שתנסה להצליח לסווג בצורה טובה יותר מבחירות רנדומלית.

* בבניית הרשת נשתמש בפונקציית האקטיבציה Relu בכל שכבת קונבולוציה, וב-softmax על ה-output layer.
* לאחר כל שכבת קונבולוציה נבצעMax Poling בגודל 2X2

**סיכום תוצאות:**

Layer (type) Output Shape Param #

=================================================================

conv2d (Conv2D) (None, 124, 124, 128) 9728

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation (Activation) (None, 124, 124, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d (MaxPooling2D) (None, 62, 62, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_1 (Conv2D) (None, 60, 60, 128) 147584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_1 (Activation) (None, 60, 60, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_1 (MaxPooling2 (None, 30, 30, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten (Flatten) (None, 115200) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense (Dense) (None, 128) 14745728

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_2 (Activation) (None, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 100) 12900

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_3 (Activation) (None, 100) 0

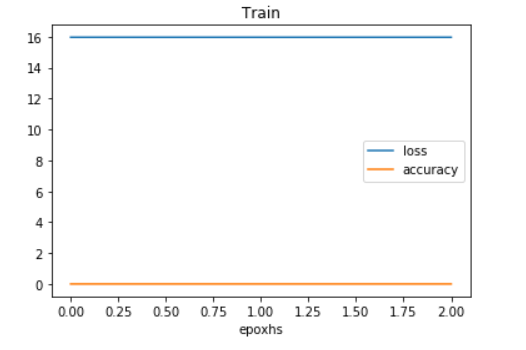
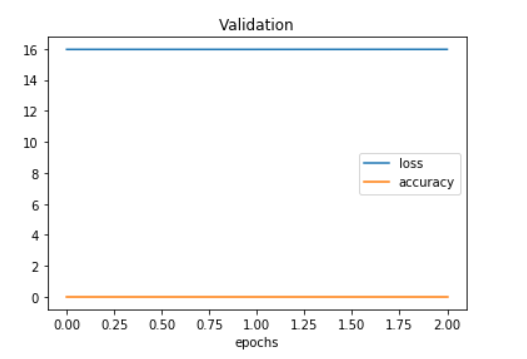
=================================================================

Total params: 14,915,940

Trainable params: 14,915,940

Non-trainable params: 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 3 | 0.9 | 10 | 0.01 | SGD - Momentum | categorical\_crossentropy |



**מסקנות:**

בניסוי זה קיבלנו את הערכים הסופיים- loss: 15.9625 - acc: 0.0097.

ניתן לראות כי האימון של הרשת לא היה מוצלח, ובניית רשת מינימלית עם Optimizer SGD לא נותן תוצאה מיטבית.

**חלק שני- שינוי ה-Optimizer:**

בניסוי הקודם ראינו שהרשת לא החזירה תוצאות רצויות, לכן נבחן האם שינוי ב- optimizer יביא תוצאות טובות.

בניסוי זה נשנה **אך ורק את ה-Optimizer** ל-Adam.

**סיכום תוצאות:**

Layer (type) Output Shape Param #

=================================================================

conv2d\_4 (Conv2D) (None, 124, 124, 64) 4864

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_8 (Activation) (None, 124, 124, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_4 (MaxPooling2 (None, 62, 62, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_5 (Conv2D) (None, 60, 60, 128) 73856

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_9 (Activation) (None, 60, 60, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_5 (MaxPooling2 (None, 30, 30, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_2 (Flatten) (None, 115200) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_4 (Dense) (None, 128) 14745728

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_10 (Activation) (None, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_5 (Dense) (None, 100) 12900

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_11 (Activation) (None, 100) 0

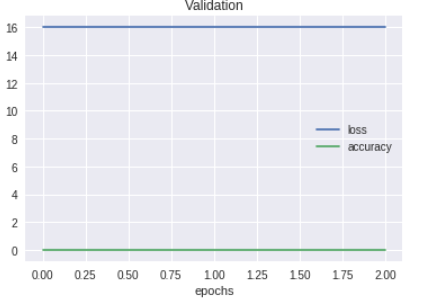
=================================================================

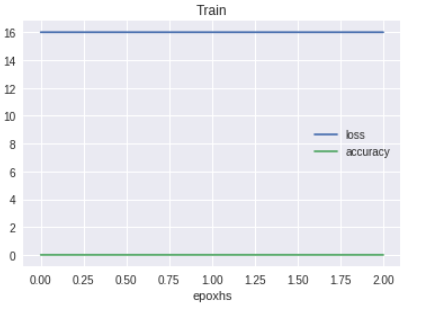
Total params: 14,837,348

Trainable params: 14,837,348

Non-trainable params: 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 3 | - | 10 | 0.01 | Adam | categorical\_crossentropy |





**מסקנות:**

בניסוי זה קיבלנו את הערכים הסופיים- loss: 16.0074 - acc: 0.0069

ניתן לראות כי האימון של הרשת לא היה מוצלח, ובניית רשת מינימלית עם Optimizer Adam נותן תוצאה פחות טובה מאשר הרשת עם SGD Optimizer.

**חלק שלישי- הגדלת ה-Batch Size:**

בניסוי הקודם ראינו שהשינוי ב-optimizer לא הביא הישג משמעותי, על כן ננסה לשנות את ה-batch size ולראות השפעתו על הרשת.

**סיכום תוצאות:**

Layer (type) Output Shape Param #

=================================================================

conv2d (Conv2D) (None, 124, 124, 128) 9728

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation (Activation) (None, 124, 124, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d (MaxPooling2D) (None, 62, 62, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_1 (Conv2D) (None, 60, 60, 128) 147584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_1 (Activation) (None, 60, 60, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_1 (MaxPooling2 (None, 30, 30, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten (Flatten) (None, 115200) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense (Dense) (None, 128) 14745728

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_2 (Activation) (None, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 100) 12900

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_3 (Activation) (None, 100) 0

=================================================================

Total params: 14,915,940

Trainable params: 14,915,940

Non-trainable params: 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 3 | 0.9 | 30 | 0.01 | SGD - Momentum | categorical\_crossentropy |



**מסקנות:**

בניסוי זה קיבלנו את הערכים הסופיים- loss: 16.0074 - acc: 0.0069

מטרת ה-batch size הינו להגדיר את מספר התמונות שעוברים עליהן באיטרצית אימון אחת (epoch) לפני עדכון המשקלים ברשת.

כנראה שהניסוי לא היה מוצלח מאחר ואין מספיק שכבות קונבולוציה לאימון הרשת (בניסוי זה הוגדרו רק 2), ואם נוסיף עוד שכבות תהיה משמעות חזקה יותר ל-batch size.

**חלק רביעי- הגדלת מספר שכבות הקונבולוציה ל-4:**

בניסוי הקודם הגענו למסקנה שהשינוי ב-Batch Size לא הניב תוצאות משמעותיות מאחר והיה מספר מינימלי של שכבות קונבולוציה (2).

לאחר חקירה באינטרנט, קראנו שעדיף להתחיל ממספר סביר של פילטרים בשכבת הקונבולוציה הראשונה על מנת לעשות "חלוקה גסה" של התמונות (**Low level features**) – מציאת מאפיינים גיאומטריים בתמונה כדוגמת קו ישר/ קו מעוקל וכד'. לאחר שכבה זו יש להגדיל את כמות הפילטרים בשכבות הקונבולוציה הבאות על מנת להגיע לחלוקה מדויקת ככל הניתן עד לסיווג ה-feature-ים עצמם (**High level features**); ובעצם למצוא כך את המאפיינים של התמונה כדוגמת הר/ נחל/ מבנה ספציפי וכד'.

יחד עם זאת, החלטנו להתחיל מגודל לא קטן יחסית עבור הפילטר הראשון, ולאורך שכבות הקונבולוציה לצמצם את גודל הפילטר. נקודת ההנחה שלנו הייתה שבהתחלה אנחנו רוצים לעשות את החלוקה הגסה, וככל שנתקדם בשכבות הקונבולוציה ברשת נרצה להגיע לרמת דיוק גבוהה יותר באמצעות מעבר על שטח קטן יותר בתמונה.

לכן, הגדלנו את מספר שכבות הקונבולוציה ברשת ל-4, כאשר הרשת בנויה כך:

* שכבה ראשונה- **64** פילטרים בגודל **16X16**, פונקציית אקטיבציה Relu, Max Polling בגודל 2X2
* שכבה שנייה- **128** פילטרים בגודל **8X8**, פונקציית אקטיבציה Relu, Max Polling בגודל 2X2
* שכבה שלישית- **256** פילטרים בגודל **4X4**, פונקציית אקטיבציה Relu, Max Polling בגודל 2X2
* שכבה רביעית- **512** פילטרים בגודל **2X2**, פונקציית אקטיבציה Relu, Max Polling בגודל 2X2

\*\* גודל הפילטרים הינו חזקות של 2, לאחר קריאה בפורומים באינטרנט שזהו סטנדרט מקובל.

בנוסף, הוספנו לרשת שכבת Dense נוספת עם 1024 פילטרים, הגדלנו את מספר ה-epoch-ים ל-5 וכן הגדלנו את גודל ה- Batch Size ל-32 (לאחר בדיקה באינטרנט כי זהו מספר סטנדרטי ומקובל, ולכן רצינו ליישר קו).

**סיכום תוצאות:**

Layer (type) Output Shape Param #

=================================================================

conv2d\_2 (Conv2D) (None, 113, 113, 64) 49216

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_4 (Activation) (None, 113, 113, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_2 (MaxPooling2 (None, 56, 56, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_3 (Conv2D) (None, 49, 49, 128) 524416

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_5 (Activation) (None, 49, 49, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_3 (MaxPooling2 (None, 24, 24, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_4 (Conv2D) (None, 21, 21, 256) 524544

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_6 (Activation) (None, 21, 21, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_4 (MaxPooling2 (None, 10, 10, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_5 (Conv2D) (None, 9, 9, 512) 524800

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_7 (Activation) (None, 9, 9, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_5 (MaxPooling2 (None, 4, 4, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_1 (Flatten) (None, 8192) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_2 (Dense) (None, 1024) 8389632

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_8 (Activation) (None, 1024) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_3 (Dense) (None, 100) 102500

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_9 (Activation) (None, 100) 0

=================================================================

Total params: 10,115,108

Trainable params: 10,115,108

Non-trainable params: 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 5 | 0.9 | 32 | 0.01 | SGD - Momentum | categorical\_crossentropy |



**מסקנות:**

הרשת לא הביאה תוצאות טובות, כנראה ששכבות הקונבולוציה לא הצליחו לסנן את ה-feature-ים ולסווג אותם בצורה טובה. יתכן ועבור 100 קטגוריות של נופים נצטרך להמשיך ולעבות את הרשת על מנת לראות שינוי.

**חלק חמישי- הגדלת מספר שכבות הקונבולוציה ל-6:**

בעקבות הניסוי הקודם, ביצענו חקירה נוספת באינטרנט על מבנה שכבות הקונבולוציה.

הגענו ל-2 עובדות מעניינות:

1. **מומלץ לבצע Max Polling לאחר 2 שכבות קונבולוציה לפחות** ולא לאחר כל שכבת קונבולוציה**-** על מנת שנוכל לבצע פעמיים את פונקציית האקטיבציה Relu וכך יתבצע שינוי משמעותי יותר על הרשת.
2. **המשמעות של פילטר בגודל 3X3 אשר רץ פעמיים ברצף** (ללא Max Polling ביניהם) נותן אפקט יעיל כמו של פילטר אחד בגודל 7X7.

בנוסף, הוספנו את הפונקציה Early Stop שלא הייתה קיימת עד כה ברשת ל-Callback של האימון, ומטרתה לעצור את האימון של הרשת במידה ואין שיפור ב-val\_acc לאחר 2 epoch-ים.

**סיכום תוצאות:**

Layer (type) Output Shape Param #

=================================================================

conv2d (Conv2D) (None, 254, 254, 64) 1792

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation (Activation) (None, 254, 254, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_1 (Conv2D) (None, 252, 252, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_1 (Activation) (None, 252, 252, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d (MaxPooling2D) (None, 126, 126, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_2 (Conv2D) (None, 124, 124, 128) 73856

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_2 (Activation) (None, 124, 124, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_3 (Conv2D) (None, 122, 122, 128) 147584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_3 (Activation) (None, 122, 122, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_1 (MaxPooling2 (None, 61, 61, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_4 (Conv2D) (None, 59, 59, 256) 295168

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_4 (Activation) (None, 59, 59, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_5 (Conv2D) (None, 57, 57, 256) 590080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_5 (Activation) (None, 57, 57, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_2 (MaxPooling2 (None, 28, 28, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten (Flatten) (None, 200704) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense (Dense) (None, 1024) 205521920

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_6 (Activation) (None, 1024) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 512) 524800

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_7 (Activation) (None, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_2 (Dense) (None, 100) 51300

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_8 (Activation) (None, 100) 0

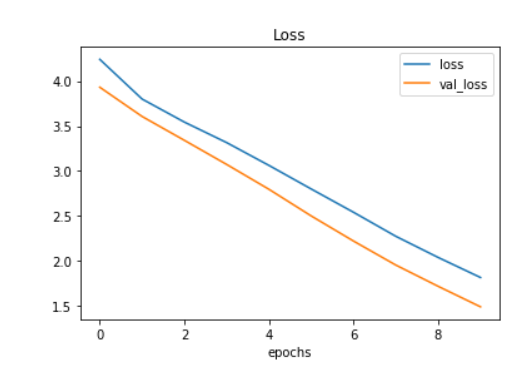
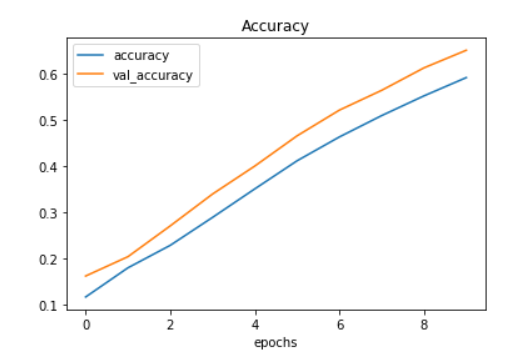
=================================================================

Total params: 207,243,428

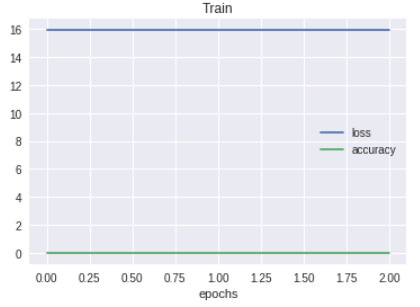
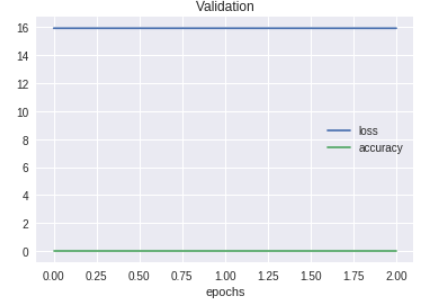
Trainable params: 207,243,428

Non-trainable params: 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 5 | 0.9 | 32 | 0.01 | SGD - Momentum | categorical\_crossentropy |

טרם האימון על 100 קטגוריות, ביצענו אימון מקדים על 5 קטגוריות על מנת לבדוק האם הרשת שלנו בכיוון הנכון.

מאחר והרשת השתפרה והיא מבצעת למידה של המידע והעלתה את הדיוק, נבצע בדיקה על 100 הקטגוריות שלנו.



**מסקנות:**

ניתן לראות כי הניסוי המצומצם היה מוצלח והביא דיוק טוב של כ- 60%.

לעומת זאת, הניסוי עם כל המידע (100 מחלקות) לא הניב תוצאה רצויה. נראה שהרשת הייתה בנויה בצורה טובה, אך היא לא הייתה בנויה למספר גדול של קטגוריות, ונראה שנדרש להוסיף עוד שכבות פילטור על מנת שהרשת תדע לסווג כמות מחלקות גדולה.

**חלק שישי- הגדלת מספר שכבות הקונבולוציה ל-7:**

בהמשך לניסוי הקודם, נרצה להוסיף שכבת קונבולוציה נוספת עם 256 פילטרים ולבדוק השפעתה על הרשת.

**סיכום תוצאות:**

Layer (type) Output Shape Param #

=================================================================

conv2d\_6 (Conv2D) (None, 254, 254, 64) 1792

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_9 (Activation) (None, 254, 254, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_7 (Conv2D) (None, 252, 252, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_10 (Activation) (None, 252, 252, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_3 (MaxPooling2 (None, 126, 126, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_8 (Conv2D) (None, 124, 124, 128) 73856

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_11 (Activation) (None, 124, 124, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_9 (Conv2D) (None, 122, 122, 128) 147584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_12 (Activation) (None, 122, 122, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_4 (MaxPooling2 (None, 61, 61, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_10 (Conv2D) (None, 59, 59, 256) 295168

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_13 (Activation) (None, 59, 59, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_11 (Conv2D) (None, 57, 57, 256) 590080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_14 (Activation) (None, 57, 57, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_12 (Conv2D) (None, 55, 55, 256) 590080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_15 (Activation) (None, 55, 55, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_5 (MaxPooling2 (None, 27, 27, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_1 (Flatten) (None, 186624) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_3 (Dense) (None, 1024) 191104000

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_16 (Activation) (None, 1024) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_4 (Dense) (None, 512) 524800

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_17 (Activation) (None, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_5 (Dense) (None, 100) 51300

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_18 (Activation) (None, 100) 0

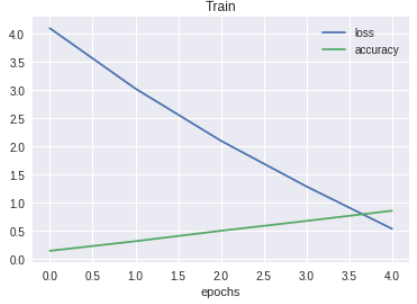
=================================================================

Total params: 193,415,588

Trainable params: 193,415,588

Non-trainable params: 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 5 | 0.9 | 32 | 0.01 | SGD - Momentum | categorical\_crossentropy |



**מסקנות:**

זאת הפעם הראשונה שהרשת הניבה תוצאות והצליחה להתאמן אך בכל זאת, עדיין הערכים לא מספיק טובים.

נראה כי המסקנה מניסוי מספר 5 הייתה נכונה, ועלינו להמשיך ולהוסיף שכבות קונבולוציה.

**חלק שביעי- הגדלת מספר שכבות הקונבולוציה ל-8:**

בהמשך לניסוי הקודם, נוסיף שכבת קונבולוציה נוספת עם 512 פילטרים ונבדוק השפעתה על הרשת.

**סיכום תוצאות:**

Layer (type) Output Shape Param #

=================================================================

conv2d\_13 (Conv2D) (None, 254, 254, 64) 1792

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_19 (Activation) (None, 254, 254, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_14 (Conv2D) (None, 252, 252, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_20 (Activation) (None, 252, 252, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_6 (MaxPooling2 (None, 126, 126, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_15 (Conv2D) (None, 124, 124, 128) 73856

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_21 (Activation) (None, 124, 124, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_16 (Conv2D) (None, 122, 122, 128) 147584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_22 (Activation) (None, 122, 122, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_7 (MaxPooling2 (None, 61, 61, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_17 (Conv2D) (None, 59, 59, 256) 295168

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_23 (Activation) (None, 59, 59, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_18 (Conv2D) (None, 57, 57, 256) 590080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_24 (Activation) (None, 57, 57, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_19 (Conv2D) (None, 55, 55, 256) 590080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_25 (Activation) (None, 55, 55, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_8 (MaxPooling2 (None, 27, 27, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_20 (Conv2D) (None, 25, 25, 512) 1180160

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_26 (Activation) (None, 25, 25, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_9 (MaxPooling2 (None, 12, 12, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_2 (Flatten) (None, 73728) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_6 (Dense) (None, 1024) 75498496

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_27 (Activation) (None, 1024) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_7 (Dense) (None, 512) 524800

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_28 (Activation) (None, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_8 (Dense) (None, 100) 51300

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_29 (Activation) (None, 100) 0

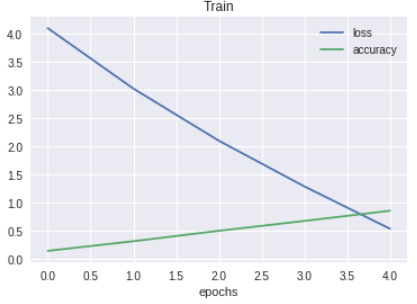
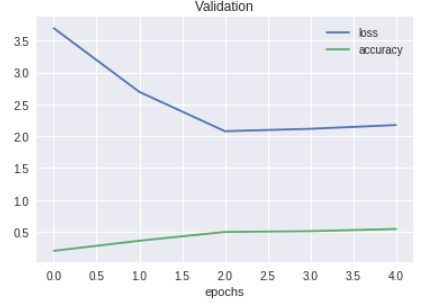
=================================================================

Total params: 78,990,244

Trainable params: 78,990,244

Non-trainable params: 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 5 | 0.9 | 32 | 0.01 | SGD - Momentum | categorical\_crossentropy |



**מסקנות:**

נראה שיש שינוי לטובה בגרף של ה-loss לעומת הניסוי הקודם, וכן שהוספת שכבת הקונבולוציה תרמה לחיוב לאימון מדויק יותר של הרשת.

**חלק שמיני- הגדלת מספר שכבות הקונבולוציה ל-10:**

בעקבות השיפור שקרה בניסוי הקודם, נגדיל את מספר שכבות הקונבולוציה ל10, כאשר כל אחת מהשכבות החדשות תהיה בעלת 512 פילטרים (ללא Max Polling ביניהן)

**סיכום תוצאות:**

Layer (type) Output Shape Param #

=================================================================

conv2d\_21 (Conv2D) (None, 254, 254, 64) 1792

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_30 (Activation) (None, 254, 254, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_22 (Conv2D) (None, 252, 252, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_31 (Activation) (None, 252, 252, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_10 (MaxPooling (None, 126, 126, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_23 (Conv2D) (None, 124, 124, 128) 73856

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_32 (Activation) (None, 124, 124, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_24 (Conv2D) (None, 122, 122, 128) 147584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_33 (Activation) (None, 122, 122, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_11 (MaxPooling (None, 61, 61, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_25 (Conv2D) (None, 59, 59, 256) 295168

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_34 (Activation) (None, 59, 59, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_26 (Conv2D) (None, 57, 57, 256) 590080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_35 (Activation) (None, 57, 57, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_27 (Conv2D) (None, 55, 55, 256) 590080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_36 (Activation) (None, 55, 55, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_12 (MaxPooling (None, 27, 27, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_28 (Conv2D) (None, 25, 25, 512) 1180160

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_37 (Activation) (None, 25, 25, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_29 (Conv2D) (None, 23, 23, 512) 2359808

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_38 (Activation) (None, 23, 23, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_30 (Conv2D) (None, 21, 21, 512) 2359808

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_39 (Activation) (None, 21, 21, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_13 (MaxPooling (None, 10, 10, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_3 (Flatten) (None, 51200) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_9 (Dense) (None, 1024) 52429824

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_40 (Activation) (None, 1024) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_10 (Dense) (None, 512) 524800

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_41 (Activation) (None, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_11 (Dense) (None, 100) 51300

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_42 (Activation) (None, 100) 0

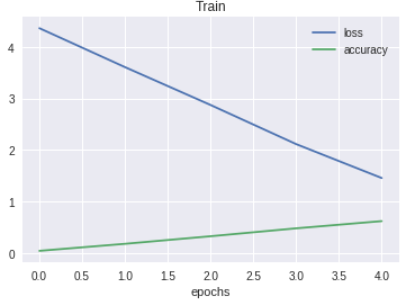
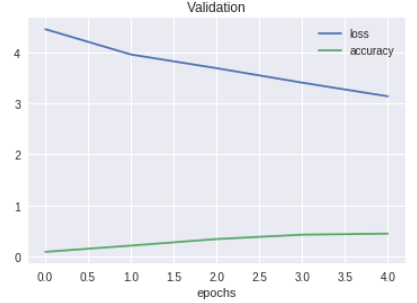
=================================================================

Total params: 60,641,188

Trainable params: 60,641,188

Non-trainable params: 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 5 | 0.9 | 32 | 0.01 | SGD - Momentum | categorical\_crossentropy |



**מסקנות:**

נראה שהוספת 2 שכבות קונבולוציה נוספות אכן תרמו לדיוק של רשת הנוירונים. עם זאת, נראה שלא בוצעו מספיק אימונים על הרשת עם המידע, וקצב הלמידה של הרשת את המידע היה איטי (הגרף ליניארי ולא פרבולה).

**חלק תשיעי- הגדלת מספר ה-Epoch-ים ל-15:**

Epoch מוגדר כמספר האימונים שנבצע על הרשת.

עד כה ביצענו כ-5 epoch-ים, משמע שבאימון אחד עברנו כ-5 פעמים על כל המידע ואימנו את הרשת עפ"י המודל שיצרנו.

הגרף בניסוי הקודם היה טוב, אך הסקנו כי קצב האימון על המידע היה איטי ולכן נרצה לבדוק את הרשת עם מספר גדול יותר של Epoch-ים.

**סיכום תוצאות:**

Layer (type) Output Shape Param #

=================================================================

conv2d\_21 (Conv2D) (None, 254, 254, 64) 1792

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_30 (Activation) (None, 254, 254, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_22 (Conv2D) (None, 252, 252, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_31 (Activation) (None, 252, 252, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_10 (MaxPooling (None, 126, 126, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_23 (Conv2D) (None, 124, 124, 128) 73856

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_32 (Activation) (None, 124, 124, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_24 (Conv2D) (None, 122, 122, 128) 147584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_33 (Activation) (None, 122, 122, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_11 (MaxPooling (None, 61, 61, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_25 (Conv2D) (None, 59, 59, 256) 295168

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_34 (Activation) (None, 59, 59, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_26 (Conv2D) (None, 57, 57, 256) 590080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_35 (Activation) (None, 57, 57, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_27 (Conv2D) (None, 55, 55, 256) 590080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_36 (Activation) (None, 55, 55, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_12 (MaxPooling (None, 27, 27, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_28 (Conv2D) (None, 25, 25, 512) 1180160

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_37 (Activation) (None, 25, 25, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_29 (Conv2D) (None, 23, 23, 512) 2359808

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_38 (Activation) (None, 23, 23, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_30 (Conv2D) (None, 21, 21, 512) 2359808

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_39 (Activation) (None, 21, 21, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_13 (MaxPooling (None, 10, 10, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_3 (Flatten) (None, 51200) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_9 (Dense) (None, 1024) 52429824

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_40 (Activation) (None, 1024) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_10 (Dense) (None, 512) 524800

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_41 (Activation) (None, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_11 (Dense) (None, 100) 51300

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_42 (Activation) (None, 100) 0

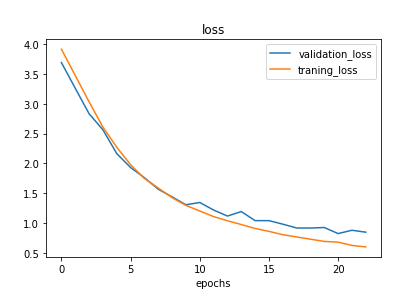
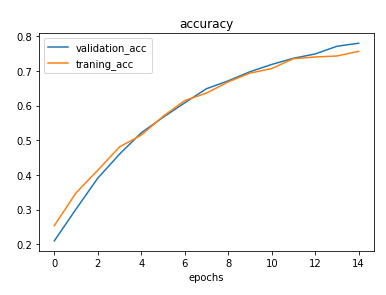
=================================================================

Total params: 60,641,188

Trainable params: 60,641,188

Non-trainable params: 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 15 | 0.9 | 32 | 0.01 | SGD - Momentum | categorical\_crossentropy |



**מסקנות:**

ניתן לראות שהגדלת מספר ה-epoch-ים ל15 **הניב תוצאות יפות מאוד!**

**הגענו לרמת דיוק של ה-validation data של כ-80%!**

**חלק עשירי- הגדלת מספר ה-Epoch-ים ל-25:**

בניסוי הקודם ראינו כי הגדלת מספר ה-epoch-ים ל-15 הניב רמת דיוק על ה-validation data של כ-80%. זהו היה שיפור לעומת חלק 8 שבו היה לנו רמת דיוק של כ-60%.

בניסוי זה נרצה לראות האם הגדלת מספר ה-epoch-ים ל-25 יגדיל יותר את val\_acc.

**סיכום תוצאות:**

Layer (type) Output Shape Param #

=================================================================

conv2d\_21 (Conv2D) (None, 254, 254, 64) 1792

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_30 (Activation) (None, 254, 254, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_22 (Conv2D) (None, 252, 252, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_31 (Activation) (None, 252, 252, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_10 (MaxPooling (None, 126, 126, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_23 (Conv2D) (None, 124, 124, 128) 73856

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_32 (Activation) (None, 124, 124, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_24 (Conv2D) (None, 122, 122, 128) 147584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_33 (Activation) (None, 122, 122, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_11 (MaxPooling (None, 61, 61, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_25 (Conv2D) (None, 59, 59, 256) 295168

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_34 (Activation) (None, 59, 59, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_26 (Conv2D) (None, 57, 57, 256) 590080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_35 (Activation) (None, 57, 57, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_27 (Conv2D) (None, 55, 55, 256) 590080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_36 (Activation) (None, 55, 55, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_12 (MaxPooling (None, 27, 27, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_28 (Conv2D) (None, 25, 25, 512) 1180160

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_37 (Activation) (None, 25, 25, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_29 (Conv2D) (None, 23, 23, 512) 2359808

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_38 (Activation) (None, 23, 23, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_30 (Conv2D) (None, 21, 21, 512) 2359808

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_39 (Activation) (None, 21, 21, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_13 (MaxPooling (None, 10, 10, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten\_3 (Flatten) (None, 51200) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_9 (Dense) (None, 1024) 52429824

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_40 (Activation) (None, 1024) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_10 (Dense) (None, 512) 524800

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_41 (Activation) (None, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_11 (Dense) (None, 100) 51300

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_42 (Activation) (None, 100) 0

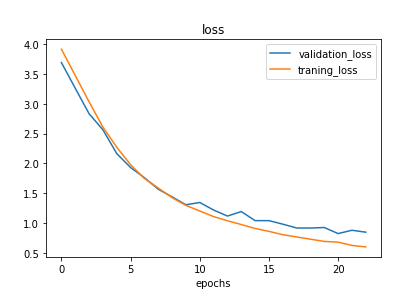
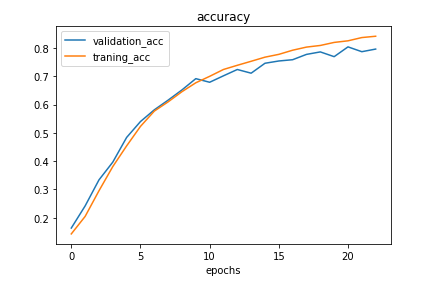
=================================================================

Total params: 60,641,188

Trainable params: 60,641,188

Non-trainable params: 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 25 | 0.9 | 32 | 0.01 | SGD - Momentum | categorical\_crossentropy |



**מסקנות:**

ניתן לראות שהגדלת מספר ה-epoch-ים ל25 לא הביאה שינוי משמעותי לרשת, וכנראה שהגדלה נוספת של ה-epoch-ים לא תשנה את val\_acc.

**חלק אחת עשרה- הוספת Batch Normalization:**

לאחר חקירה באינטרנט, החלטנו בניסוי זה להוסיף לרשת שלנו את שכבת Batch Normalization.

שכבה זו תקטין את הערכים בכל המשקולות באופן פרופורציונלי, במטרה להפוך את החישובים למהירים יותר.

**סיכום תוצאות:**

Layer (type) Output Shape Param #

=================================================================

conv2d\_1 (Conv2D) (None, 254, 254, 64) 1792

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_v1 (Batc (None, 254, 254, 64) 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation (Activation) (None, 254, 254, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_2 (Conv2D) (None, 252, 252, 64) 36928

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_v1\_1 (Ba (None, 252, 252, 64) 256

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_1 (Activation) (None, 252, 252, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d (MaxPooling2D) (None, 126, 126, 64) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_3 (Conv2D) (None, 124, 124, 128) 73856

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_v1\_2 (Ba (None, 124, 124, 128) 512

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_2 (Activation) (None, 124, 124, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_4 (Conv2D) (None, 122, 122, 128) 147584

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_v1\_3 (Ba (None, 122, 122, 128) 512

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_3 (Activation) (None, 122, 122, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_1 (MaxPooling2 (None, 61, 61, 128) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_5 (Conv2D) (None, 59, 59, 256) 295168

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_v1\_4 (Ba (None, 59, 59, 256) 1024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_4 (Activation) (None, 59, 59, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_6 (Conv2D) (None, 57, 57, 256) 590080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_v1\_5 (Ba (None, 57, 57, 256) 1024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_5 (Activation) (None, 57, 57, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_7 (Conv2D) (None, 55, 55, 256) 590080

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_v1\_6 (Ba (None, 55, 55, 256) 1024

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_6 (Activation) (None, 55, 55, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_2 (MaxPooling2 (None, 27, 27, 256) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_8 (Conv2D) (None, 25, 25, 512) 1180160

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_v1\_7 (Ba (None, 25, 25, 512) 2048

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_7 (Activation) (None, 25, 25, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_9 (Conv2D) (None, 23, 23, 512) 2359808

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_v1\_8 (Ba (None, 23, 23, 512) 2048

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_8 (Activation) (None, 23, 23, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_10 (Conv2D) (None, 21, 21, 512) 2359808

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_v1\_9 (Ba (None, 21, 21, 512) 2048

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_9 (Activation) (None, 21, 21, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_3 (MaxPooling2 (None, 10, 10, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

flatten (Flatten) (None, 51200) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense (Dense) (None, 1024) 52429824

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_v1\_10 (B (None, 1024) 4096

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_10 (Activation) (None, 1024) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_1 (Dense) (None, 512) 524800

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

batch\_normalization\_v1\_11 (B (None, 512) 2048

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_11 (Activation) (None, 512) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_2 (Dense) (None, 100) 51300

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

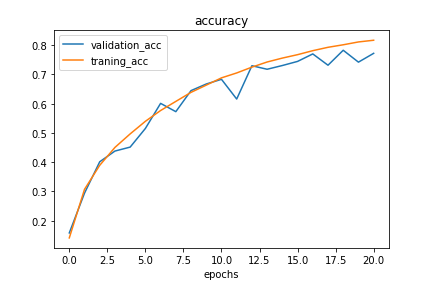
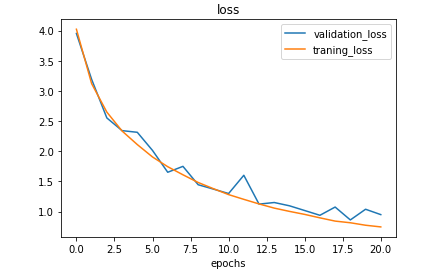
batch\_normalization\_v1\_12 (B (None, 100) 400

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

activation\_12 (Activation) (None, 100) 0

=================================================================

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Epochs | momentum | Batch Size | Learning Rate | Optimizer | Loss Function |
| 25 | 0.9 | 32 | 0.01 | SGD - Momentum | categorical\_crossentropy |



**מסקנות:**

נראה שהוספת Batch Normalization על כל אחת מהשכבות לא השפיעה במידה רבה על הרשת.

**חלק שתיים עשרה- הוספת Dropout Layer:**

שכבה זו לוקחת סט רנדומלי של משקלים ברשת ומאפסת אותן, וזאת בשביל למנוע מצב של Overfitting.

מאחר ובכל הניסויים שלנו לא התרחש מצב של overfitting, לא הוספנו את שכבה זו.

במידה והיה נדרש- היינו מוסיפים בהתאם לאורך שכבות הקונבולוציה וה-Dense עם ערך סביר (20%-30%).

**סיווג ע"י KNN**

בחלק האחרון של הפרויקט, לקחנו את הרשת הטובה ביותר שהצלחנו לייצר ולאחר עיבוד הכנסנו את הPredictions יחד עם קבוצת התיוגים למודל KNN, כך שבכל פעם שנתבקש למצוא את עשרת השכנים הקרובים ביותר של תמונה חדשה כלשהיא, נוכל להשתמש בKNN בכדי למצוא את עשרת הדומות לה ביותר, בהתבסס על פרדיקציה של המודל הנבחר.

באופן זה ניבחן את יכולות המודל שלנו על ה-test\_data ונראה האם הצלחנו במשימת האופטימיזציה שהוטלה עלינו.

את התהליך ניתן לראות בשלמותו תחת הפרויקט ב-github.