**קונבולוציה ברשת נוירונים לסיווג תמונות**

**וautoencoders- לריטוש תמונות**

גבריאל מיכאלי

בהנחיית ד"ר מיה הרמן

**תוכן עניינים**

**1.מבוא:**

-למידה עמוקה

-רשתות נוירונים(כולל רקע ביולוגי)

-מבנה רשתות נוירונים

#-למידה (back propagation feedforward and)

-יתרונות וחסרונות של רשת נוירונים

-שימושים ברשתות נוירונים

**2.רשתות נוירונים קונבולוציות (CNN):**

-רשת CNN הסבר

-מוטיבציה לשימוש ב-CNN

-קלט ופלט הרשת

- מבנה רשת קונבולוציה

-למידת תכונות הקלט ומניעת overfitting(RGB,stride, kernel,filter)

-Classification

- #בניית ועיצוב ה-Dataset

-שימוש בפונקציית אופטימיזציה ושגיאה

-אימון הרשת על תמונות

-תוצאות רשת CNN על תמונות

- ResNET

- דוגמאות למאגרי נתונים נפוצים המשתמשים בCNN

**3.מבוא לautoencoders**

-מהו autoencoder

-מבנה ה-autoencoder (encoder,decoder)

-קונבולוציה ב-autoencoder

-שימושים של autoencoders

- שימוש הautoencoder- לריטוש תמונות

-אימון autoencoder לריטוש תמונות

-פונקציית שגיאה ב-autoencoder

-תוצאות הautoencoder- על תמונות

-דוגמא לשימוש בautoencoder- לתמונות מהספרות

**סיכום**

**ביבליוגרפיה**

תוכן עניינים-(יסודר יותר בהמשך)

[**1.מבוא** 4](#_Toc138342708)

[**1.1 למידה עמוקה** 4](#_Toc138342709)

[**1.2 רשתות נוירונים-רקע ורקע ביולוגי** 6](#_Toc138342710)

[**1.2.1 פרספטרון** 7](#_Toc138342711)

[**מבנה רשתות נוירונים וסוגים** 8](#_Toc138342712)

[**יתרונות וחסרונות של רשת נוירונים** 8](#_Toc138342713)

[**שימושים ברשתות נוירונים** 9](#_Toc138342714)

[**רשתות נוירונים קונבולוציות (CNN)** 9](#_Toc138342715)

[**רשת CNN הסבר** 9](#_Toc138342716)

[**מוטיבציה לשימוש ב-CNN** 10](#_Toc138342717)

[**קלט ופלט הרשת** 10](#_Toc138342718)

[**מבנה רשת קונבולוציה** 11](#_Toc138342719)

[**למידת תכונות הקלט** 13](#_Toc138342720)

[**Classification ברשתות** 14](#_Toc138342721)

[**שימוש בפונקציית אופטימיזציה ושגיאה** 15](#_Toc138342722)

[**בניית ואימון הרשת על תמונות** 17](#_Toc138342723)

[**תוצאות רשת CNN על תמונות** 18](#_Toc138342724)

[**resNET** 18](#_Toc138342725)

[**דוגמאות למאגרי נתונים נפוצים המשתמשים בCNN** 20](#_Toc138342726)

[**3. מבוא לautoencoders** 20](#_Toc138342727)

[**3.1 מהו autoencoder (מקודד אוטומטי)** 21](#_Toc138342728)

[**מבנה ה-autoencoder (encoder,decoder)** 21](#_Toc138342729)

[**3.2 שימושים של autoencoders** 22](#_Toc138342730)

[File Compression (דחיסת קבצים)- 22](#_Toc138342731)

[Image transformation (טרנספורמציה של תמונה)- 22](#_Toc138342732)

[Image De-noising (ריטוש תמונה) 22](#_Toc138342733)

[**שימוש הautoencoder- לריטוש תמונות** 22](#_Toc138342734)

[**חילוץ נתונים ב-autoencoder לריטוש תמונות** 23](#_Toc138342735)

[**אימון ופונקציית שגיאה בautoencoder** 25](#_Toc138342736)

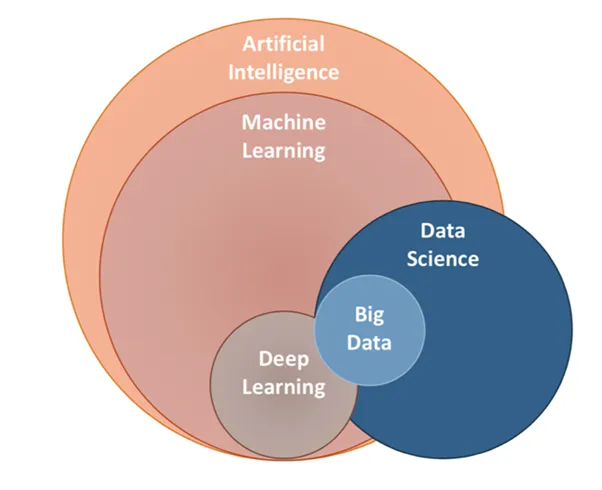
[ביבליוגרפיה 26](#_Toc138342737)

# **1.מבוא**

## **1.1 למידה עמוקה**

למידה עמוקה היא טכניקת למידת מכונה שמלמדת מחשבים לעשות את מה שבא טבעי לבני אדם, שזה למידה מדוגמאות. למידה עמוקה היא טכנולוגיית מפתח מאחורי מכוניות ללא נהג, המאפשרת להן לזהות תמרור עצור, או להבחין בין הולך רגל לעמוד תאורה. זהו המפתח לשליטה קולית במכשירים צרכניים כמו טלפונים, טאבלטים, טלוויזיות ורמקולים דיבוריים. למידה עמוקה זוכה לתשומת לב רבה לאחרונה ולא סתם. בזכות למידה עמוקה אנו יכולים להשיג תוצאות שלא היו אפשריות קודם לכן בלי היכולת המפותחת של למידה עמוקה.

בלמידה עמוקה, מודל ממוחשב לומד לבצע משימות סיווג ישירות מתמונות, טקסט או קול. מודלים של למידה עמוקה יכולים להגיע לדיוק מתקדם, שאף לפעמים עולים על הביצועים של האדם. מודלים אלו מאומנים באמצעות קבוצה גדולה של נתונים מסומנים (לייבל) וארכיטקטורות רשתות עצביות המכילות שכבות רבות.

למידה עמוקה היא צורה מיוחדת של למידת מכונה, כלומר היא חלק ממנה , כפי שלמידת מכונה היא חלק מבינה מלאכותית , בדיוק כפי שניתן לראות בתמונה מספר 1:

איור 1 ההיררכייה בתחום בינה מלאכותית[21]

מה שאנו יכולים להבין מהתמונה הזו הוא שבינה מלאכותית (AI) הוא החלק העיקרי כלומר הוא התחום הרחב , למידת מכונה (ML) היא קבוצת המשנה של AI ולמידה עמוקה(DL) היא תת קבוצת של ML. לפיכך AI הוא חלק רחב יותר הכולל ML ו-DL.

כעת כדי להבין טיפה יותר טוב מה כל חלק אומר נסביר את המונחים:

בינה מלאכותית, היא מכונה שלומדת לחשוב כמו בני אדם. AI מתייחס לסימולציה של אינטליגנציה אנושית במכונות שמתוכנתות לחשוב כמו בני אדם ולחקות את פעולותיהם.

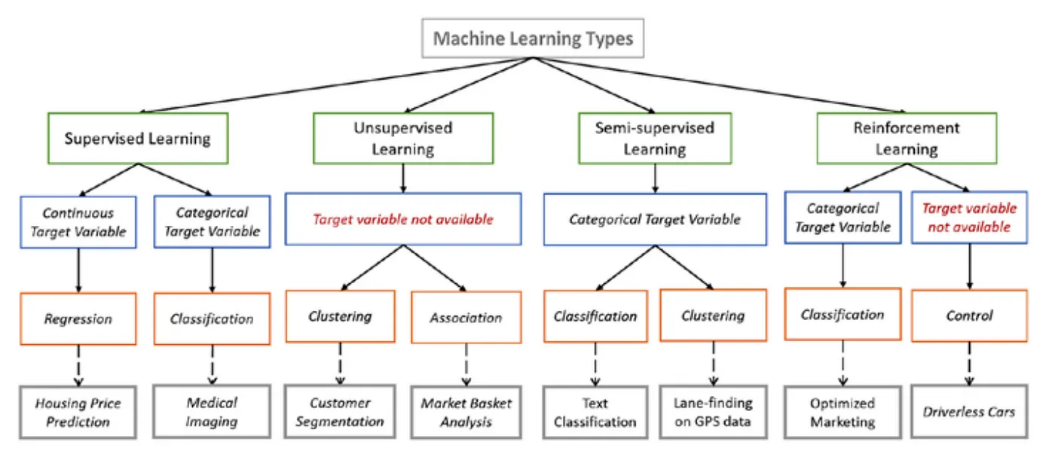
למידת מכונה, הוא יישום של בינה מלאכותית (AI) המספק למערכות את היכולת ללמוד ולהשתפר מניסיון באופן אוטומטי מבלי להיות מתוכנתים במפורש. לפיכך, ML, באופן כללי, מייצרת מכונה כדי ללמוד מהנתונים הנתונים.

למידה עמוקה , גורם למכונה ללמוד נתון נתון לעומק. למידה עמוקה היא תת-קבוצה של למידת מכונה בבינה מלאכותית (AI) שיש לה רשתות המסוגלות ללמוד את עצמן מהנתונים על ידי יצירת רשת עצבית, בדומה לרשת הקיימת במוח שלנו.

עכשיו כפי שיש סוגים שונים של בינה מלאכותית גם יש סוגים שונים של למידה עמוקה, יש למידה מפוקחת שנקראתsupervied learning , יש חצי-מפוקחת שנקראת semi-supervied ובלתי מפוקחת שנקראת learning Unsupervised, בנוסף, יש סוג נוסף הנקרא למידה מחיזוקים כלומר Reinforcement learning.

אלגוריתמי למידה מפוקחים, מנסים ליצור מודל של קשרים ותלות בין פלט חיזוי היעד לתכונות הקלט, כך שנוכל לחזות את ערכי הפלט עבור נתונים חדשים בהתבסס על אותם קשרים שהוא למד ממערכי הנתונים הקודמים.

אלגוריתמי למידה בלתי מפוקחת, הן אלגוריתמים בהם המחשב מאומן עם נתונים ללא תווית. כלומר אין לנו כאן איזה כאן תווית אשר מורה על הנתונים, למעשה המחשב עשוי ללמד אותנו דברים חדשים לאחר שהוא לומד דפוסים בנתונים, ולרוב אלגוריתמים אלה מנסים להשתמש בטכניקות על נתוני הקלט כדי לכרות חוקים, לזהות דפוסים, ולסכם ולקבץ את נקודות הנתונים המסייעות בהסקת תובנות משמעותיות ומתארות את הנתונים טוב יותר למשתמשים.



איור 2: תיאור גישות בלמידה עמוקה22]]

אלגוריתמי למידה חצי מפוקחת, הם אלגוריתמים אשר מתאימים ביותר בהיעדר תוויות ברוב התצפיות אך נוכחות במספר מועט. לעיתים ישנם מצבים שבהם השמה של תווית על הנתונים זה משימה מאוד יקרה ולכן אנו נשים תווית רק על חלק מהנתונים ולכן במצב כזה למידה חצי מפוקחת מתאימה, שיטה זו מנצלת את הרעיון שלמרות שהחברות בקבוצה של הנתונים הלא מסומנים אינם ידועים, נתונים אלו נושאים מידע חשוב על פרמטרי הקבוצה.

אלגוריתמי למידה מחיזוקים, הם אלגוריתמים המאפשרים למכונות ולסוכני תוכנה לקבוע באופן אוטומטי את ההתנהגות האידיאלית בהקשר מסוים, על מנת למקסם את הביצועים שלה. נדרש משוב פשוט לתגמול כדי שהסוכן ילמד את התנהגותו; זה ידוע בתור אות החיזוק, כך שכאשר יש התנהגות נכונה של הסוכן/ מודל אנו נתגמל אותו ולהפך.

בנוסף גם בנושא של סיווג תמונות יש לנו אלגוריתמים מלמידה עמוקה שמתאימים ובאלגוריתמים אלה למידה עמוקה חוותה הצלחה רבה. כל תמונה יכולה להכיל מידע רב כך שלעיתים יש קושי לעבור על כולו, בלמידה עמוקה ישנם שיטות כמו רשת CNN(שעליה נסביר יותר בהמשך) אשר משתמש בשיטה של למידה מפוקחת ובה אנו ממטבים את הרשת כך שאנו נוכל להוציא ממנו את הפיצרים החשובים ביותר מכל תמונה, כך שאנו נוכל בצורה פשוטה יותר לקבוע את הסיווג של התמונה למספר קבוצות, ובדרך זו למשל אנו יכולים לפשט את הבעיה של סיווג תמונות ולחסוך במשאבים.

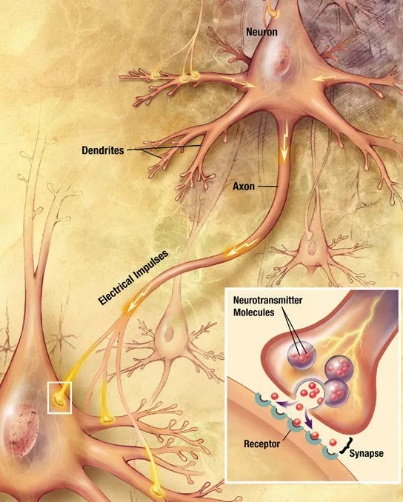
לכן ללמידה עמוקה יש הצלחה רבה בתחום הסיווג תמונה ובעוד תחומים רבים אחרים , ומזה אפשר להבין למה למידה עמוקה זוכה להעדה רבה והצלחה רבה כך שהיא כל פעם נלמדת ומתפתחת יותר וגם בזמן כתיבת שורות אלו מתבצעים שידרוגים והתפתחויות חדשות בתחום.

## **1.2 רשתות נוירונים-רקע ורקע ביולוגי**

נוירון מלאכותי דומה לנוירון ביולוגי, שנהם מקבלים קלט מהנוירונים האחרים, מבצעים עיבוד מסוים ומייצר פלט. רשתות נוירונים ביולוגים השפיעו רבות על רשתות נוירונים מלאכותיות, רשתות נוירונים ביולוגיים ומלאכותיות בנויים שנהם מנוירונים.

נוירון הוא תא במוח האנושי שאחראי על קבלת קלט חושי מהעולם החיצוני, שליחת אותות לשרירים אנושיים, והפיכת האותות החשמליים והעברתם בכל שלב ביניהם.

נוירון מורכב מ-3 חלקים עיקריים, הדנדריט, האקסון וגוף התא. הדנדריט הוא המקום שבו נוירון מקבל קלט מנוירונים אחרים. האקסון הוא הפלט של נוירון שהוא מעביר את האות לנוירונים אחרים. גוף התא מכיל גרעין וחומר גנטי, השולטים בפעילות התא. הדרך שבה הנוירונים מתקשרים זה עם זה היא באמצעות אותות אשר הם מעביריים אחד לשני הנקראים נוירוטרנסמיטורים והם עוברים ע"י חלל צר הנקרא סינפסה, שנמצא בין האקסון של הנוירון השולח לבין ל-לדנדריטים.



איור 3:תיאור תא עצב[23]

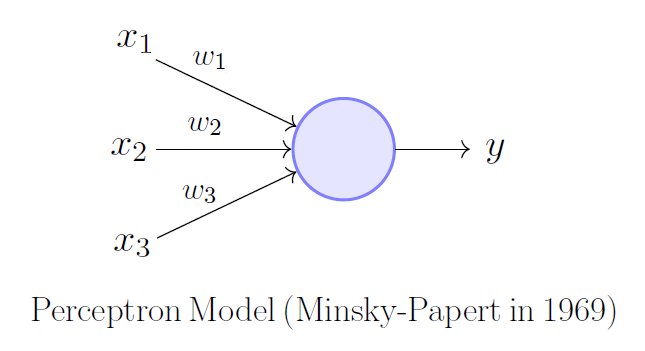
כפי שהסברנו בהתחלה נוירון מלאכותי דומה מאוד לנוירון טבעי, וניתן לראות את זה במבנה של הנוירון המלאכותי שמזכיר את הטבעי . נוירון מלאכותי מורכב מ-3 חלקים עיקריים, משקל והטיה כדנדריט המסומן ב-w ו-b בהתאמה, פלט כאקסון מסומן ב-y, ותפקוד ההפעלה כגוף תא שמסומן על ידי f(x) , כך שה-x הוא אותות הקלט שמתקבלים על ידי הדנדריט. נשים לב שבנוירונים מלאכותיים, קלט ומשקל מיוצגים כווקטור ואילו הטיה מיוצגת כסקלר. נוירון מלאכותי מעבד אותות קלט על ידי ביצוע מכפלה פנימית בין וקטור הקלט לווקטור המשקל, ולזה הוא מוסיף את איבר ההטיה, ולאחר מכן הפעלת פונקציית הפעלה(עליה נסביר בהמשך), ולבסוף מפיץ את התוצאה לנוירונים אחרים.

אך כיצד הכלה ההתפתחות של הרשת נוירונים?

### **1.2.1 פרספטרון**

הפרספטרון הוא הרשת העצבית העתיקה ביותר, שנוצרה כל הדרך בשנת 1958. זוהי גם הרשת העצבית הפשוטה ביותר. פותח על ידי פרנק רוזנבלט, הפרספטרון היווה את הבסיס ליסודות הרשתות העצביות.

פרספטרון הינה רשת עצבית המכילה רק נוירון אחד, מה שהופך אותה לפשוטה ביותר. כלומר הפרספטרון לוקח דגימות ומכפיל אותם במשקלים מתאימים, כך שהוא מחשב רק פלט אחד. חסרון הפספטרון הוא שהוא סובל בגלל חוסר המורכבות שלו בכך שהוא יכול לעבד נתונים רק ברמת מורכבות אחת , כלומר בגלל שמכיל רק נוירון אחד.



איור 4: מבנה הפספטרון[2]

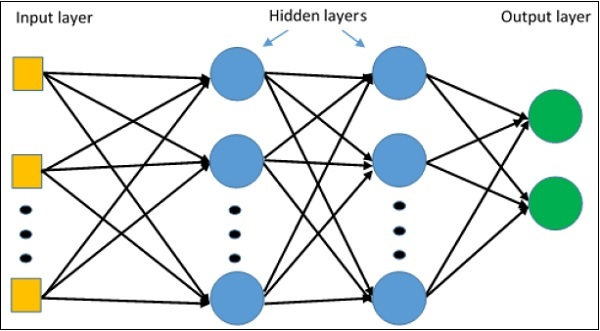
התפתחות מעניינת שנוצרה עם הפספטרון הוא ה-פרספטרון רב-שכבתי, כלומר multi-level-precpectron (MLP) , זה עדיין פרספטרון אולם ישנה מורכבות נוספת בעקבות הופעתן של שכבות שהן: שכבת הקלט (input layer) , שכבה נסתרת (hidden layers) ושכבת הפלט (output layers).

ששכבת הקלט מכילה את הנתונים שלנו (מקבלת קלט מספרי ולכן יתכן ומניפולציה על הקלט תהיה רצויה על מנת להתאים את הקלט)

שכבות חבויות הן, שכבות המורכבות מרוב הנוירונים ברשת העצבית והן מה שמהוות הלב של מניפולציה של הנתונים כדי לקבל פלט רצוי. הנתונים יעברו דרך השכבות הנסתרות ויטופלו על ידי משקלים והטיות רבות. שמה (שכבות חבויות) ניתן לה על מנת הלא יוכלו לבצע בה שינויים בניגוד לשכבת הקלט והפלט,

שכבת הפלט היא התוצאה של הנתונים כשהם עוברים דרך הרשת העצבית, והמטרה אליה אנו מנסים להגיע.

התפתחות זו בעצם היא הבסיס לרשת נוירונים , מכוון ש-MLP היא רשת נוירונים מלאה (כפי שניתן לראות בתמונה 4) , ולאחר מכן החלו התפתחויות והיה ניתן להגיע להסרה של חלק מהקשרים.



איור 5: מבנה ה-MLP ,רשת נוירונים מלאה[2]

## **מבנה רשתות נוירונים וסוגים**

כפי ש-MLP מורכב מ3 שכבות גם רשת הנוירונים מורכבת מ3 שכבות שהן בדיוק כמו ב-MLP, שכבת הקלט, שכבות חבויות/נסתרות, שכבת הפלט , שהן גם מתפקדים בדיוק כמו ב-MLP.

כל רשת נוירונים יכולה להיות מורכבת ממספר שונה של שכבות נסתרות ומספר שונה של נוירונים , כך שבחירת מבנה של הרשת תשפיע ותקבע בסופו של דבר את התוצאות שיתקבלו, אך בשונה מ-MLP פה כל שכבה ברשת לא חייבת להיות מחוברת במלואה לשכבה הבאה, דבר זה יכול לגרום למצב של תוצאות משופרות ומתוך רצון לא להגיע להתאמת יתר או כדי לחסוך במשאבים מיותרים שיכולים לחסוך גם זמן.

כפי שהסברנו בפרק של הרקע ורקע ביולוגי: "נוירון מלאכותי מעבד אותות קלט על ידי ביצוע מכפלה פנימית בין וקטור הקלט לווקטור המשקל, ולזה הוא מוסיף את איבר ההטיה, ולאחר מכן הפעלת פונקציית הפעלה(עליה נסביר בהמשך), ולבסוף מפיץ את התוצאה לנוירונים אחרים"

אך מזה בעצם פונקציית הפעלה (activation function)?

פונקציות הפעלה היא פונקציה שנוירון מלאכותי משתמש בה כדי לקבל את הפלט של נוירון, היא ידועה גם כ-Transfer Function. התוצאה של המכפלה הפנימית בין משקל לקלט פלוס הטיה היא בטווח של (מינוס אינסוף, אינסוף) ולכן פונקציית ההפעלה שואפת למפות את התוצאה לטווח מסוים בהתאם לפונקציה. פונקציה חשובה אחת היא פונקציית ה-הסיגמואידית (sigmoid) , שהיא באמת יכולה למפות את הטווח מ(מינוס אינסוף, אינסוף) ל(0,1) כך שהוא מייצג את ההסתברות של x אם x שייך למחלקה 1 או 0, כלומר אם x מקבל לאחר הפוקנציה ערך שגדול מ0.5 אז x מסווג כ-1 , אחרת יסווג כ-0.

## **יתרונות וחסרונות של רשת נוירונים**

כמו לכל דבר גם לרשת נוירונים יש את היתרונות וחסרונות עליהם נעבור כעת.

אחד היתרונות של רשתות עצביות מתפקדות היטב עם נתונים ליניאריים ולא ליניאריים, אך ביקורת נפוצה על רשתות עצביות, היא שהן דורשות מגוון גדול של הכשרה לפעולה בעולם האמיתי. הסיבה לכך היא שכל מכונת למידה צריכה מספיק דוגמאות מייצגות כדי ללכוד את המבנה הבסיסי שמאפשר לה להכליל למקרים חדשים , דבר זה יכול לגרום לבעיה כאשר יש לנו בעיה שאנו רוצים לפתור אך אין לנו מספיק נתונים כדי ללמד את הרשת שלנו.

בנוסף יש לרשת נוירונים יתרון בכך שאם ישנם כמה יחידות מתוך הרשת שלא עובדות , כלומר לא מצליחות להגיב אז הרשת עדיין יכולה לעבוד , אך נשים לב כאשר מדובר ברשתות גדולות במיוחד צריך משאבים של איחסון וגיבוי מאוד גדולים דבר שיכול להוות בעיה.

בנוסף אנו צריכים להבין שרשת נוירונים יכולה ללמוד הרבה נתונים מנותחים אך אין אנו יכולים לדעת איך באמת הרשת עושה זאת , כלומר רשת נוירונים היא סוג של קופסא שחורה, והן מספקות מעט מאוד מידע ותובנות לגבי מה המודלים האלו עושים.

## **שימושים ברשתות נוירונים**

לרשתות נוירונים יש שימושים רבים בתחום הלמידת מכונה , ולכל שימוש יש את הסוג רשת נוירונים אשר מתאימה לו. נעבור כעת על חלק מהשימושים והרשתות המתאימות להם.

אחד השימושים ברשתות נוירונים הן עיבוד שפה טבעית אשר בה להבין את השפה של בני אדם בכתיבה ודיבור, בנוסף גם ניתן להשתמש ברשת על מנת לחזות את שוק המניות , שני אלה הם סוג של משימות אשר מתאימות לרשת שנקראת RNN, שזו רשת שלוקחת בחשבון שהנתונים שהיא מקבלת תלויים במופעי העבר שלהם על מנת לחזות את העתיד שלהם, והיא עושה זאת על ידי אחסון הפלט האחרון בזכרון שלו .

בנוסף אנו יכולים להשתמש ברשת על מנת לסווג תמונות, או לזהות תבניות בתמונות, שזהו משימה אשר מתאימה לרשת מסוג CNN שמשתמשת בקונבולוציה ברשת שלה . בנוסף יש גם משימות שאנו

נרצה בעצם לצמצם את התצוגה של הנתונים שלו , ובמקרים כאלה נשתמש גם בקונבולוציה אך הפעם נשתמש בה ב-autoencoders. עבור שני סוגי הרשתות אללו אנו נרחיב יותר בהמשך ונסביר עליהם יותר לעומק ועל התפקיד של כל אחד מהם.

# **רשתות נוירונים קונבולוציות (CNN)**

## **רשת CNN הסבר**

אחד מן התחומים הרבים של למידת מכונה, הוא תחום ראייה ממוחשבת. מטרת התחום הזה היא לספק למכונות את האפשרות לראות את העולם כפי שבני האדם רואים, כלומר לפשט את את המידע שלנו לצורה פשוטה לעיבוד , כך שאנו שומרים על המידע/מאפיינים חשובים ושמזה נוכל להסיק מידע עבור מספר רב של משימות, ולכן מזה בה רשתות קונבולוציה שלה יש שימושים רבים כגון: זיהוי תמונה ווידאו, ניתוח וסיווג תמונה, פנאי מדיה, המלצה מערכות, עיבוד שפה טבעית וכו'.נשים לב שבכל אחד מהתחומים האלו נעשתה התקדמות רבה לאורך השנים וזה הרבה בזכות אלגוריתמים מתקדמים שכוללים בעיקר אלגוריתם אחד מסוים שהוא רשת קונבולוצייה ((CNN.

בתחום התמונות שאנו עוסקים בו, לרשתות קונבולוציה יש את היכולות לקלוט תמונת קלט, להקצות חשיבות (משקלים והטיות ניתנות ללמידה) להיבטים/אובייקטים שונים בתמונה, ולהיות מסוגל להבדיל אחד מהשני. בנוסף לרשתות קונבולוציה יתרון נוסף שהוא כמות המשאבים הקטנה שנדרש לעיבוד מקדים לעומת שאר אלגוריתמים של סיווג.

כשמו , רשת קונבולוציה בעצם משתמש בנוסחת הקונבולוציה (שעליה הסברנו במבוא) שזה הדבר שכה מייחד אותה , בעזרתה אנו נעשה את הטרנספורמציות השונות על תמונת הקלט שלנו , להוריד את המימד הקלט שלנו ולהוציא מהקלט את הפיצרים החשובים והאינפורמטיבים ביותר , דבר אשר מספק לנו התאמה מאוד טובה לסיווג תמונות אשר מכילים מספר רב של פיקסלים שמתוכם נוכל לקחת את החשובים ביותר ולפשט/ להקטין את כמות הקלט שלנו.

אך שאלה הנשאלת היא למה לא פשוט להשתמש בטכניקה של רשתות ללא קונבולוציה במקום ברשת עם קונבולוציה לסיווג תמונות, כלומר למה שפשוט לא נשתמש אפילו באלגוריתמים אחרים של למידת מכונה לסווג את התמונות שלנו?

על זה נרחיב יותר וניתן את התשובה כעת בכותרות מוטיבציה לשימוש ב-CNN

## **מוטיבציה לשימוש ב-CNN**

מה המוטיבציה שלנו לשימוש ברשת CNN ? מדוע לא להשתמש באלגוריתמים שונים ופשוטים יותר?

ראשית נבין שתמונה אינה אלא מטריצה ​​של ערכי פיקסלים, אז נשאלת השאלה למה לא פשוט לשטח את התמונה, כלומר להפוך אותה לקלט ממימד אחד (לדוגמא: 3X3 להפוך ל-9X1) ולכניס אותה ל-Multi-Level Perceptron (שעליה הסברנו שמבוסס אלגוריתם של רשת נוריונים) על מנת שנוכל לסווג בפשטות יותר?[10]

התשובה לכך היא שישנם מקרים של תמונות בינאריות בסיסיות ביותר בהן השיטה של לשטח תמונה עשויה להביא תוצאות דיוק ממוצעות לצורכי חיזוי של מחלקות וישנם מקרים מסובכים יותר שבהן הדיוק יהיה מועט עד ללא דיוק בכלל כאשר מדובר בתמונות מורכבות בעלות תלות בפיקסלים לאורך כל הדרך.

מוטיבציה נוספת לשימוש ברשת קונבולוציה היא המסוגלות ללכוד בהצלחה את התלות המרחבית והזמנית בתמונה באמצעות יישום מסננים רלוונטיים. הארכיטקטורה מבצעת התאמה טובה יותר

למערך הנתונים של התמונה בשל הפחתת מספר הפרמטרים המעורבים והאפשרות לשימוש חוזר של משקלים. כלומר, ניתן לאמן את הרשת להבין טוב יותר את התחכום של התמונה.[10]

## **קלט ופלט הרשת**

קלט רשת קונבולוציה הוא מערך של 4D . לנתוני קלט יש צורה של (batch\_size, height, width, depth), כאשר הממד הראשון מייצג את ה-batch של התמונה שזה כמות מידע אשר מעברים בכל פעם ושלושת הממדים האחרים מייצגים ממדים של התמונה שהם גובה, רוחב

ועומק. את עומק התמונה ניתן להסביר כמספר ערוצי הצבע. לדוגמה, לתמונת RGB תהיה עומק של 3, ולתמונה בגווני אפור יהיה עומק של 1.[12]

פלט הרשת תלוי במטרתנו , זה יכול להיות סיווג של תמונה לקבוצה מסויימת וזה יכול גם להיות מערך של 4D אשר גם יכיל (batch\_size, height, width, depth) כך שה-batch ישאר אותו הדבר ושלושת המימדים האחרים יכולים להשתנות בהתאם לרשת שלנו.

## **מבנה רשת קונבולוציה**

ארכיטקטורה של רשתות CNN טיפוסיות כוללת בדרך כלל את החלפה בין קונבולוציה לאיסוף שכבות. ארכיטקטורות ה-CNN מורכבות מחמישה חלקים:

-שכבת הקלט

- שכבת הקונבולציה

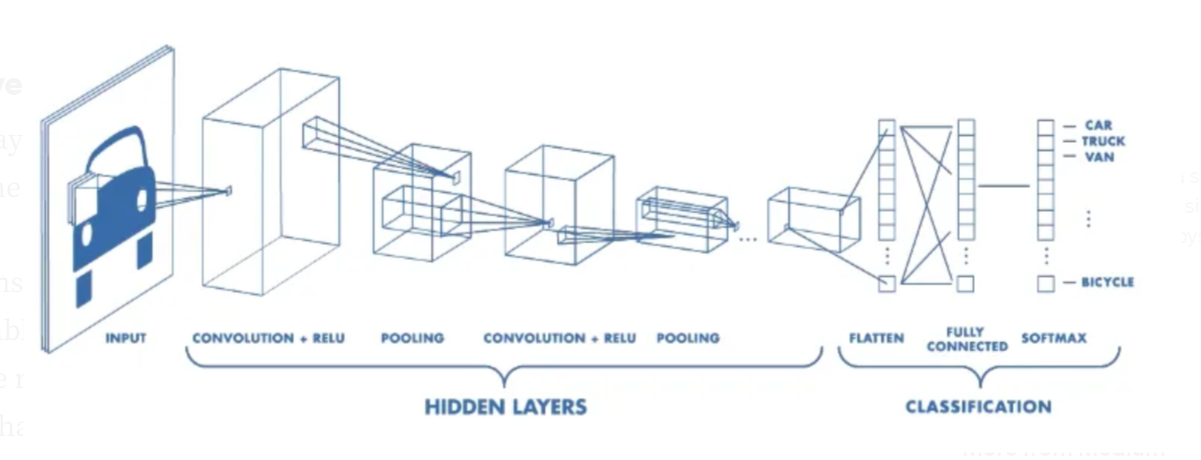
- שכבת ה-pooling,

-שכבת Fully Connected Layer

- ושכבת הפלט.

לארכיטקטורות של CNN יש תפקיד חשוב בתכנון ארכיטקטורות רשתות עצביות, שכן ארכיטקטורת רשת סבירה יותר יכולה לשפר את אפקט ההתאמה בין שכבות או להפחית חישובים מיותרים ברשת, מה שבדרך כלל מסמל שהיא יכולה להביא לביצועים טובים יותר.[11]

דוגמא לרשת כזו ניתן לראות באיור מספר 1 שבה יש ראשית את שכבת הקלט , לאחר מכן קונבולוציה ואז ביצוע של polling ושוב עד כמובן כמות השכבות שנרצה, ולבסוף את יש שכבת ה-fully connected.



איור מספר 6 - ארכיקטרטורת רשת CNN[11]

**כעת נסביר על כל שכבה ומה התפקיד שלה:**

**שכבת הקונבולוציה:**

שכבת הקונבולציה היא אבן הבניין המרכזית של CNN. הוא נושא את החלק העיקרי של העומס החישובי של הרשת. תפקידן העיקרי של שכבות אלה הוא לזהות מאפיינים החוזרים על עצמם בתוך קבוצת הנתונים ולמפות את המופע שלהם במפת המאפיינים (features map), לשם כך הן משתמשות בפונקציות קונבולוציה.

שכבה זו מבצעת מכפלה פנימית (שזה חישוב הקונבולוציה) בין שתי מטריצות, כאשר מטריצה ​​אחת היא קבוצת הפרמטרים הניתנים ללמידה הידועה לנו (הדאטא שלנו), והמטריצה ​​השנייה היא החלק המוגבל של השדה הקולט כלומר ה-kernel/filter. כלומר הקרנל הוא מרחב קטנן יותר מבחינה מרחבית מתמונה אבל מעמיקה יותר. המשמעות היא שאם התמונה מורכבת משלושה ערוצים (RGB), הגובה והרוחב של הקרנל יהיו קטנים מבחינה מרחבית, אך העומק משתרע עד לכל שלושת הערוצים.

הקרנל "מחליק " על התמונה ובמהלך המעבר הקרנל מחליק על פני הגובה והרוחב של התמונה ומייצר את ייצוג התמונה של אותו אזור קליטה. זה מייצר ייצוג דו מימדי של התמונה המכונה מפת האקטיבציה שנותנת את התגובה של הקרנל בכל מיקום מרחבי של התמונה.

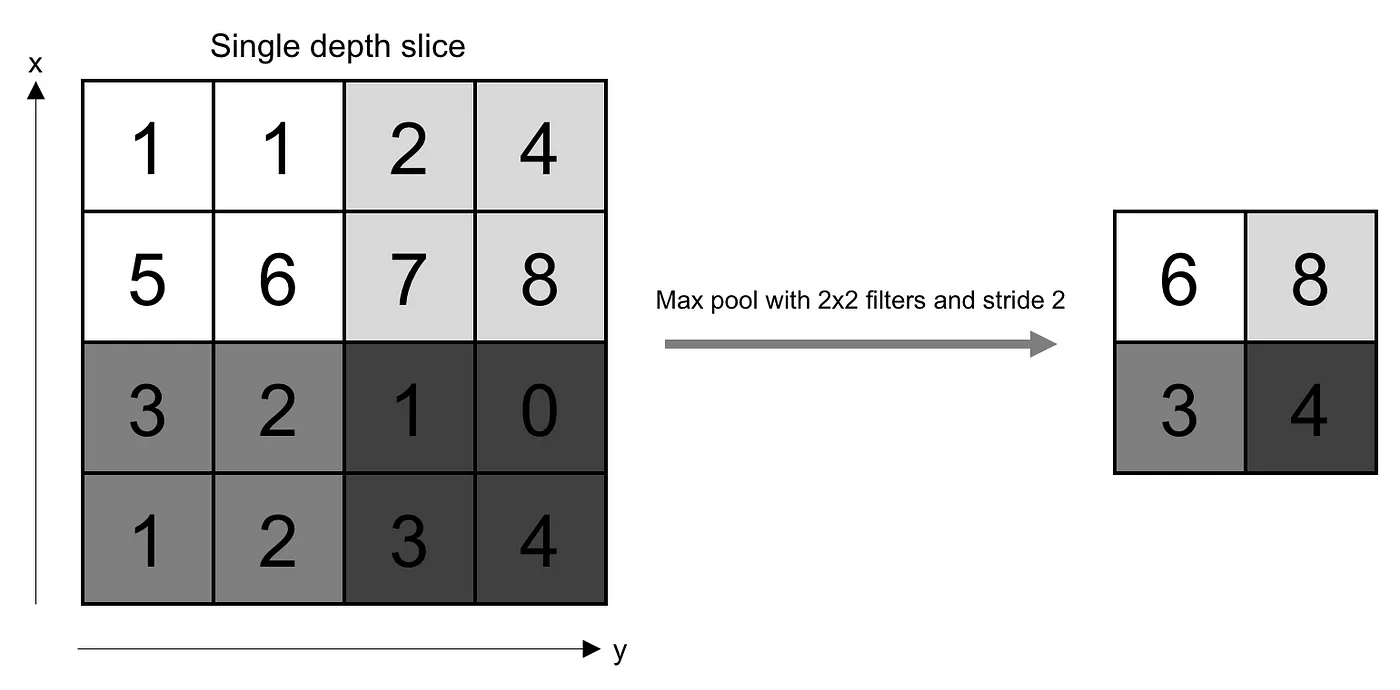
כלומר בכל מעבר של הקרנל אנו מקבלים את התכונות החשובות ביותר מכל תמונה ומקבלים את זה ביצוג דו מימדי.חשוב לזכור שהערכים שבפילטר נשארים קבועים לכל אורך הסריקה.

פרמטר נוסף שחשוב הוא ה-stride שזה בעצם גודל הפסיעה, גודל התזוזה על התמונה, כלומר הוא מתייחס לגודל הפיקסלים שאנו עוברים עליהם בכל פעולה של הפילטר. אם נניח שגודל הפסיעה הוא 1 אז הפילטר יזוז כל פעם פיקסל אחד , ואם הוא יהיה 2 אז הפילטר יזוז כל פעם בשתי פיקסלים. בהרבה מן המקרים ניתן להניח שלפיקסלים קרובים הם באותו מקוםן מרחבי , כלומר האינפורמציה שלהם זהה, ולכן בחלק מן המקרים ניתן לדלג על פיקסלים קרובים. חשוב להבין שככל שהפסיעה שלנו גדולה יותר כך גודל מפת האקטיבציה קטן יותר (כי מדלגים יותר פיקסלים) וגם אנו יכולים לשנות את הפסיעה שלנו על ציר X ו-Y עם ערכים שונים.

**שכבת ה-pooling**

שכבת ה-pooling מחליפה את הפלט של הרשת במקומות מסוימים על ידי גזירת סטטיסטיקה מסכם של הפלטים הקרובים. זה עוזר להקטין את הגודל המרחבי של הייצוג, מה שמקטין את הכמות הנדרשת של חישוב ומשקולות. פעולת האיגום מעובדת על כל פרוסה של הייצוג בנפרד.[11]

**כפי שניתן לראות בתמונה** מספר 2 אנו מבצעים גזירה סטטיסטית אשר מקינה את היצוג וכך מקטינה בעצם את כמות החישוב הנדרשת ומוציאה את הפלטים החשובים ביותר.

****

איור 7 -הדגמה של פעולת ה-Polling **[2]**

**Fully Connected Layer:**

לנוירונים בשכבה זו יש קישוריות מלאה עם כל הנוירונים בשכבה הקודמת והאחריה, שכבת ה-FC עוזרת למפות את הייצוג בין הקלט לפלט כאשר מחזירים את תוצאת הסיווג., כלומר שכבה זו היא אחראית על המיון/סיווג.

שני המסווגים הנפוצים ביותר ב-רשת קונבולוציה הם Softmax ו-SNM.

Softmax מחזיר את ההסתברויות לשיוך לכל סוג (סך ההסתברויות הם 1), ו- SVM, מחזיר את ציוני הסיווג ומתייחס לסוג שציונו הגבוה ביותר כאל הסוג הנכון.

## **למידת תכונות הקלט**

לאחר שאנו לומדים את תכונות הקלט, אנו צריכים למפות ולייצג את הפלט בצורה הטובה ביותר.

כפי שהוסבר אנו עושים זאת על ידי הוספת שכבה מחוברת מלאה שלומדת ומסווגת את הקלט שלנו, שכבה זו מקבלים וקטור עמודות שמתקבל לאחר מעבר על שכבות הקונבולוציה שבו הנתונים יושמו בשיטת feed-forword ולאחר מכן מתקיים תהליך ה-backpropagation.

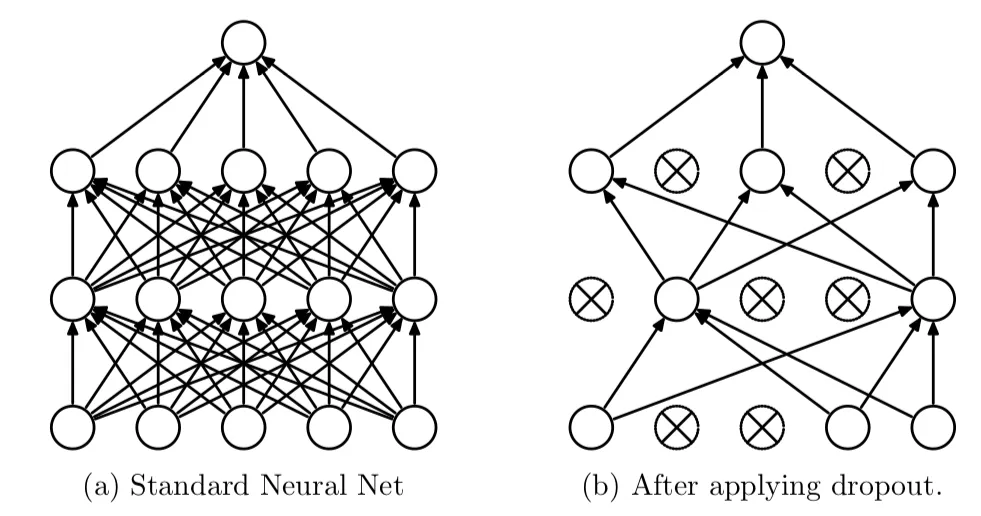
שיטת ה- feed-forword היא המעבר על הרשת אשר מבצעת את המעבר על כל שכבה שמבצעת את ההכפלה במשקולות של כל שכבה וחישוב פונקצייה בכל שכבה.

שכבת ה- backpropagation מבצעת את תהליך עידכון בכל שכבה אשר מטרתה זה לצמצם את השיגאה שיש ברשת על מנת לתת תוצאות טובות יותר.

כך בעצם הרשת לומדת ומנתחת את הקלט שאנו מקבלים .

אך בהרבה מן המקרים יכול להתקיים מצב של התאמת יתר (overfitting) שזו מצב שבו הרשת מתאימה את עצמה יתר על מידה למאגר ספציפי של נתונים, שזה דבר שמוביל להכללה נמוכה ולתוצאות/ דיוק נמוך יותר של הרשת.

על מנת למנוע זאת אנו יכולים להשתמש ב-dropout שהיא סוג של רגוליזציה [4] ,כלומר זה תהליך שבו הרשת מנתקת אקראית חלק מהנוירונים ברשת ונוירונים אלה לא ישתתפו בתהליך האימון, כך מוכל למנוע את התאמת היתר ולקבל תוצאות טובות יותר.

****כפי שניתן לראות בתמונה מספר 3, לאחר תהליך ה-dropout חלק מהקשרים בין הנוירונים מתנתקים מהרשת וכך ניתן למנוע התאמת יתר.

איור 8- דוגמא לפעולת ה-Dropout על רשת **[4]**

## **Classification ברשתות**

אחד מהנושאים החושבים בתחום של למידת מכונה זה תחום הסיווג (Classification), שזה בעצם היכולות לשייך קלט שאנו מספקים לאחד מן המחלקות שמתאימות ביותר לקלט שנתנו.

בבעיית הסיווג עבור רשתות קונבולוציה (CNN) אנו בעצם לומדים את המרכיבים החשובים ביותר שיש בקלט שלנו ואנו לפיהם מנסים לסווג את הקלט שלנו למחלקה שמתאימה ביותר בשבילו.

במקרה של סיווג תמונות , אנו מסווגים את התמונות לפי חשיבות הפיקסלים של התמונה. כל תמונה מורכבת מפיקסלים אשר מייצגים אותה, וישנם פיקסלים עם חשיבות רבה , ופיקסלים עם חשיבות מועטה. לכן בעת הסיווג של התמונה למחלקה מסוימת אנו לומדים מהם הפיקסלים החשובים ביותר ולפיהן אנו מסווגים את התמונה שלנו למחלקה הנכונה.

כפי שהסברנו ברשת קונבולוציה (CNN) , השכבות האחרונות של הרשת הן שכבות ה- Fully Connected שבהם אנו עושים את התהליך הסיווג. בשכבות אלו אנו מביאים לידי ביטוי את טכניקת ה-softmax. טכניקה זו ממירה וקטור נתון לווקטור עם ערכים חיובים שסוכמם מסתכם ל-1 והם בעצם מהווים הסתברויות. הסתברויות אלו הן אלה שבסופו של דבר יקבעו את המחלקה שאליה תסווג הקלט (התמונה) שלנו. ערך נמוך שווה להסתברות נמוכה, ערך גבוהה בוקטור שווה להסתברות גבוהה. רגרסיית Softmax היא צורה של רגרסיה לוגיסטית המנרמלת ערך קלט לווקטור של ערכים העוקב אחר התפלגות הסתברות שסך הכל שלה מסתכם ב-1. ערכי הפלט נמצאים בין הטווח [0,1] והם אלו הן אלה שבסופו של דבר יקבעו את המחלקה שאליה תסווג הקלט (התמונה) שלנו. ערך נמוך שווה להסתברות נמוכה, ערך גבוהה בוקטור שווה להסתברות גבוהה. בזכות דבר זה אנחנו מסוגלים לעשות יותר מסיווג בינארי ולהכיל כמה שיותר מחלקות ברשת שלנו.[13]

מכל זאת נוכל להסיק שעל מנת לסווג תמונות למחלקות מסוימות אנו יכולים להשתמש בפונקציית הsoftmax על מנת לקבל את ההסתברויות שהתמונה שייכת למחלקה מסוימת.

## **שימוש בפונקציית אופטימיזציה ושגיאה**

בעת בניית הרשת הנוירונים אנו צריכים לבחור את פונקציית האופטימיזציה שלנו ופונקציית השגיאה, שלהם הכרעה גדולה מאוד על ביצועי המודל שלנו .

איך ראשית מהם פונקציית שגיאה ופונקציית אופטימיזציה?

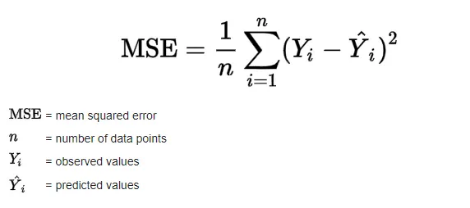
פונקציית שגיאה הינה שיטה להערכת עד כמה האלגוריתם שלך עובד טוב על מערך הנתונים שלנו. כלומר זוהי פונקציה מתמטית שבאה להראות האם אלגוריתם הלמידה שלנו הצליח על מערך הנתונים שלנו ואם כן באיזה מידה.

אופטימיזציה הוא אלגוריתם או פונקציה שמתאימים את התכונות של הרשת העצבית, כמו קצב למידה ומשקולות. לפיכך, הוא מסייע בשיפור הדיוק ומפחית את ההפסד הכולל. בנוסף פונקציית האופטיציזציה משפיעה באופן משמעותי על הדיוק של מודל הלמידה העמוקה בכך שהוא משפיע על מהירות אימון המודל שלנו.

בעת בחירה פונקציית שגיאה ואופטימיזציה אנו צריכים לחשוב טוב על סוג העבודה שלנו ועל מטרת הרשת שלנו, ומה אנו מנסים להשיג. כעת נסתכל על סוגי פונקציית השגיאה שיש ונתאים כל אחד מהם למטרה שלו:

**Mean Squared Error**:

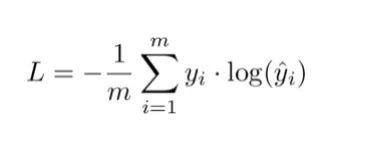
פונקציית MSE משמשת למשימות רגרסיה. כפי שניתן להבין מהשם, הפסד מחושב על ידי לקיחת הממוצע של ההבדלים בריבוע בין ערכים בפועל לערכים החזויים.

הנוסחא :

וכפי שניתן באמת להבין הנוסחא מייצגת לנו את המרחק בין הערך שמקבלים לבין הערך האמיתי.

**Binary Cross Entropy:**

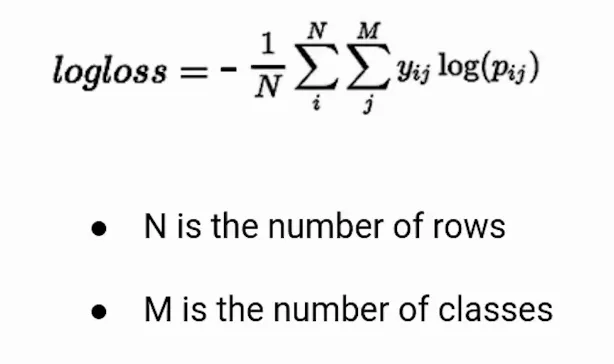
BCE היא פונקציית ההפסד המוגדרת כברירת מחדל המשמשת עבור משימות הסיווג הבינארי. זה דורש שכבת פלט אחת כדי לסווג את הנתונים לשתי מחלקות וטווח הפלט הוא אפס עד אחד, כלומר צריך להשתמש בפונקציה sigmoid אשר תנרמל את הערכים לטווח זה, שזה גם דבר הנחוץ בעת שימוש ב-BCE כי אחרת לא תעבוד כראוי.

נוסחא:

כאשר y הוא התווית בפועל,ו-ŷ הוא התפלגויות ההסתברות החזויות של המסווג לחיזוי מחלקה אחת ו-m הוא מספר הרשומות.

**Categorical Cross Entropy:**

זוהי פונקציית האובדן המוגדרת כברירת מחדל כאשר יש לנו משימת סיווג מרובת מחלקות. פונקציה זו דורשת את אותו מספר מחלקות שיש בפלט לאחר שימוש של softmax, כלומר פוקנצייה זו מקבלת את אותו מספר מחלקות לפי כמות המחלקות סיווג שיש ומקבלת את טווח הסיכויים של כל מחלקה לאחר שמבצעים softmax.

****נוסחא:

כאשר y הוא התווית בפועל ו-p הוא התפלגויות ההסתברות החזויות של המסווגן לניבוי המחלקה j.

כעת נסתכל על כמה מפונקציית האקטיבציה שיש ונתאים כל אחד מהם למטרה שלו:

**Gradient Descent (GD)**

זהו המייעל הבסיסי ביותר שמשתמש ישירות בנגזרת של פונקציית ההפסד וקצב הלמידה כדי להפחית את ההפסד ולהשיג את המינימום שגיאה שיש. גישה זו מאומצת גם בהפצה לאחור ברשתות עצביות שבהן הפרמטרים המעודכנים משותפים בין שכבות שונות בהתאם למועד השגת ההפסד המינימלי כלומר כל המשקולות מתעדכנות כאשר כל שיפוע מערך הנתונים מחושב, שזה גם דבר שמאט את התהליך.

**Stochastic Gradient Descent**

זוהי גרסה שונה של שיטת GD, שבה פרמטרי המודל מתעדכנים בכל איטרציה. זה אומר שלאחר כל מדגם אימון, פונקציית ההפסד נבדקת והמודל מתעדכן. עדכונים תכופים אלה מביאים להתכנסות למינימום בפחות זמן, אך זה בא במחיר של שונות מוגברת שיכולה לגרום לדגם לחרוג מהמיקום הנדרש.

אבל יתרון של טכניקה זו הוא דרישת זיכרון נמוכה בהשוואה לקודמתה מכיוון שכעת אין צורך לאחסן את הערכים הקודמים של פונקציות האובדן.

**RMSProp:**

מטרתו היא להפחית את האגרסיביות של קצב הלמידה על ידי לקיחת ממוצע מעריכי של השיפועים במקום הסכום המצטבר של שיפועים בריבוע.

משתמש בסימן השיפוע כדי לשנות את גודל הצעד באופן דיסקרטי עבור כל משקל. באלגוריתם זה משווים תחילה שני שיפועים על סמך הסימנים כך שאפ הסימנים שלהם זהים, תמשיך בכיוון הנכון וכך גם גודל הצעד גדל בכמות קטנה. אבל אם הסימנים שלהם הפוכים, אתה צריך להקטין את גודל הצעד. לאחר מכן, אתה מגביל את גודל הצעד וממשיך בעדכון המשקל.

**Adam**

משלב את העוצמה של RMSProp ו-GD. ב-שיטה זו הכוח של המומנטום GD להחזיק את היסטוריית העדכונים ואת קצב הלמידה האדפטיבי שמספק RMSProp הופך את adam לשיטה רבת עוצמה.

## **בניית ואימון הרשת על תמונות**

כאשר אנו באים לאמן רשת CNN על תמונות עלינו תמיד לדאוג לדאטא שישמש לאימון הרשת ולדאטא שישמש לבדיקת תוצאות הרשת שלנו. עלינו תמיד לזכור שכל המידע חייב להלקח מאותו התפלגות, כלומר הסט אימון וסט מבחן שלנו חייב להיות מאותו הדאטא.

לאחר חלוקת האימון לסט אימון וסט מבחן אנו יכולים להגדיר את הפרמטרין עבור הרשת שלנו , ועלינו לזכור שסט המבחן שלנו אף פעם לא יוכל להכלל באימון שלנו, אחרת נקבל תוצאות שגויות.

כאשר אנו רוצים לאמן רשת CNN כך שיסווג תמונות עינו לבחור את הפרמטרים הנכונים עבור הנתונים שלנו. לדוגמא כאשר אנו רוצים לסווג את התמונות לכמה מחלקות עלינו כמובן להשתמש בפונקציית שגיאה אשר תתאים לזה , לכן אנו נשתמש בפונקציית השגיאה של Categorical Cross Entropy אשר לוקחת בחשבון את זה שיש לנו כמה מחלקות והיא תסווג נכונה את הנתונים שלנו וכפי שהסברנו תשתמש ב- softmax על מנת לחשב את הסיכויים של כל מחלקה לסיווג.

בנוסף אנו צריכים לבחור גם אופטימייזר , אך במקרה זה אין לנו משהו ייחודי שמתאים במיוחד לרשת CNN , ולכן ברוב המקרים נרצה להשתמש בAdam- אך הדבר הרצוי לעשות הוא ניסוי וטעיה על מנת לראות מה הוא הנכון בשביל המודל שלנו.

דבר נוסף וחשוב הוא לדעת את מספר שכבות הקלט והפלט ואת מספר הנוירונים שלהן. לכל רשת יש שכבת קלט אחת ושכבת פלט אחת. מספר הנוירונים בשכבת הקלט שווה למספר משתני הקלט בנתונים המעובדים. מספר הנוירונים בשכבת הפלט שווה למספר הפלטים הקשורים לכל קלט.

אבל האתגר האמיתי באימון ובניית הרשת הוא לדעת את מספר השכבות הנסתרות ואת הנוירונים שלהן. הנה כמה קווים מנחים לדעת את מספר השכבות והנוירונים הנסתרים לכל שכבה נסתרת בבעיית סיווג:

ראשית עלינו לזכור שמספר הנוירונים הנסתרים בכל שכבה נסתרת חדשה שווה למספר החיבורים שיש ליצור. בנוסף כדי לחבר את הקווים שנוצרו על ידי השכבה הקודמת, מתווספת שכבה נסתרת חדשה. שימו לב ששכבה נסתרת חדשה מתווספת בכל פעם שאתם צריכים ליצור חיבורים בין השורות בשכבה הנסתרת הקודמת. דבר חשוב שחשוב לזכור זה שאלינו ליצור גבול ברור להפרדה בין המחלקות שלנו, כלומר בסופו של דבר עלינו לחזור למצב של הפרדה מדוייקת בפלט שלנו למחלקות.

כל אלה דברים שיכולים לעזור לנו וכמובן שעלינו כל פעם לבדוק אם תוספת של רמות תשפר את הביצועים של המודל שלנו.

רק לאחר שאנו עושים את כל זה עלינו להתחיל לאמן את המודל שלנו לפי פונקציית השגיאה והאופטימייזר כך שעלינו לבחור את כמות הפעמים שאנו עוברים על הרשת שלנו על מנת לאמן אותה וכך גם אנו נוכל להסתכל על התקדמות הדיוק של המודל כדי לקבוע אם אנחנו מרוצים מהאימון .

# **תוצאות רשת CNN על תמונות**

לאחר אימון הרשת על סט תמונות האימון שלנו אנו נבחן את תוצאות הרשת שלנו על סט המבחן שלנו. אנו נרצה כמובן את האחוז הדיוק המרבי ביותר.

נשים לב שלאחר שאנו נקבל את תוצאות הרשת שלנו על סט המבחן אנו תמיד נוכל לבדוק על ידי הוספה/ הסרה של שכבות אם נקבל תוצאות טובות יותר או על ידי הוספה. הסרה של כמות נוירונים בכל שכבה, כך שלאחר בדיקה זו אנו נוכל לקחת את הרשת שמביאה לנו את התוצאות הטובות ביותר ולאמן את כל הדאטא שלנו על רשת זו כך שתוכל לחזות טוב יותר מידע עתידי שנביא לו . עלינו לזכור שככל שכמות הדאטא שלנו גדולה יותר כך הרשת שלנו לומדת יותר ויש לה סיכויי דיוק גבוהים יותר עבור דאטא עתידי.

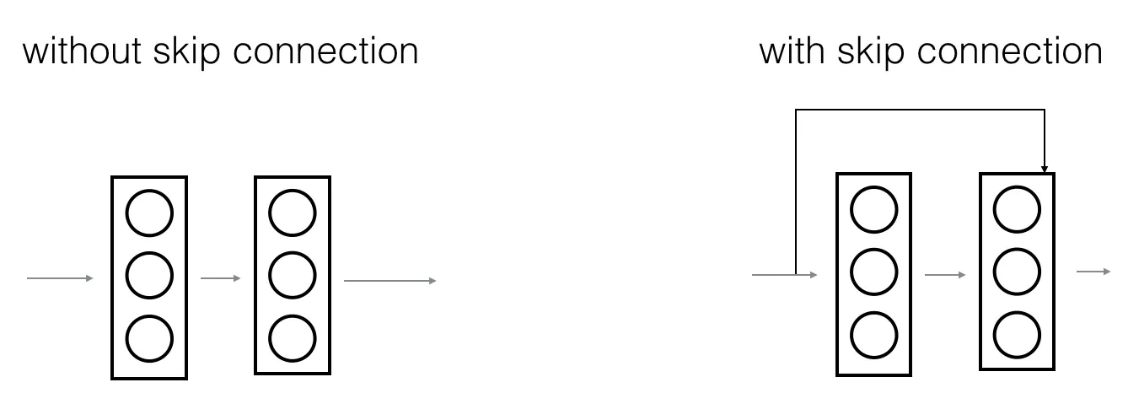
נזכירשהסברנו שכל סיווג של תמונה נעשית על ידי פונקציית ה-SOFTMAX אשר תביא בסופו של דבר את הסיווג עם ההסתברות הגדולה ביותר לכל תמונה.

## **resNET**

resNET (ראשי תיבות של (Residual Neural network הוא עוד סוג של רשת CNN.

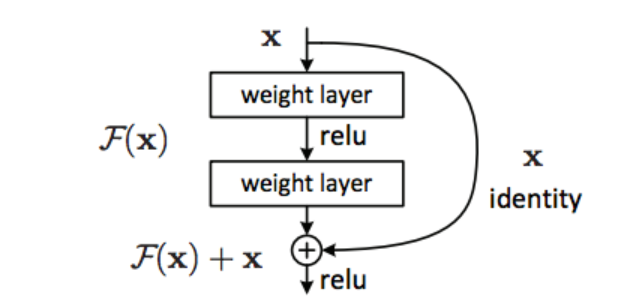
ארכיטקטורת resNET בה להתמודד עם בעיה שכיחה ברשתות נוירונים שהיא לימוד שכבות מרובות של תכונות מבלי להיתקע במינימום מקומי.

מספר שכבות של רשת משפיע באופן ישיר על ביצועי הרשת, ישנם מספר שכבות שאנו צריכיס על מנת להביע את הרשת להתכנסות ומה שקורה זה שביצועי הרשת מפסיקים לעלות , ואף אם אנו נוסיף שכבות מעבר והרשת תתעמק יותר אנו נתרחק מההתכנסות וביצועי הרשת יכולים אף לרדת. לכן אנו צריכים לחשוב מה יכול לספק לנו מהו כמות השכבות הנכונה בשבילנו כדי להגיע לעומק הנכון עבור הרשת שלנו. ResNET מספק לנו פתרון לזה.

ככל שרשת עמוקה יותר הביצועיים שלה מתחילים להדרדר במהירות גובהה יותר[18], לכן אם היינו יכולים לדלג על חלק מהשכבות כך שנוכל להגיע לדיוק בשכבות הנמוכות זה היה מצויין. זה בדיוק מה שresNET- עושה. resNET הציגה לראשונה את הרעיון של דילוג שכבות . ניתן לראות שתרשים הבא מה הכוונה שדילוג שכבות:

איור 9 דוגמא לדילוג שכבות ב-resnet[18]

אנו רואים כי בחלק הימיני של התמונה אנו עוברים רמה רמה ברשת לעומת בחלק השמאלי של התמוה בה אנו עושים דילוג ובעצם מעבירים את הקלט שלנו לשכבה הבאה.

לכן הרעיון המרכזי של resNET דילוג על קשרים (Skip-Connection) והמבנה שלה הוא כפי שניתן לראות באיור הבא:

איור 10 בלוק שיורי ב-resNET [19]

כלומר ניתן להבין שהרעיון הוא שבמקום שהשכבות ילמדו את פונקציית המיפוי המקורית (שנגדירה כ-H(x)) הרשת תתאים לערך F(x) = H(x) -x כלומר. H(x) = F(x)+x

כפי שמוצג באיור במקום לחשב את H(x) בצורה רגילה ResNet, מחשבת את פונקציית השארית שהיא F(x) ומוסיפה אליו את הפלט x וכך מקבלים את H(x). כך בניגוד לרשתות אחרות resNET מסוגל להעביר את המידע שלו ל2-3 שכבות ממנו.

לכן ה-Skip Connection הוא הרחבה מאוד מעניינת ל- CNN כי הוא מגביר את הביצועים במשימות של image classification על ידי כך שאם יש שכבה שפוגעת בביצועים של הרשת, יתבצע דילוג עליה. ובנוסף באמצעות בלוקים שיוריים ניתן לבנות רשתות עמוקות במיוחד.

## **דוגמאות למאגרי נתונים נפוצים המשתמשים בCNN**

כפי שהסברנו לרשת CNN שימושים רבים בתחום התמונות , ויש לרשת זו יכולות לפתור הרבה בעיות שקשורות לתחום הזה. אנו כעת נביא דוגמא לחלק מן המאגרי מידע הפופולריים ביותר שעושים שימוש ברשת CNN [24]:

**MNIST**- אחד ממערכי הנתונים הפופולריים ביותר של למידה עמוקה בחוץ. זהו מערך נתונים של ספרות בכתב יד ומכיל מערך הדרכה של 60,000 דוגמאות ומערך מבחן של 10,000 דוגמאות,אשר מטרתו זה להשתמש ברשת CNN על מנת לסווג את כל תמונה למספר שרשום בה.

תמונה שמכילה צילום מסך, ריבוע, גופן, מלבן

התיאור נוצר באופן אוטומטי

איור 11 : דוגמא לתמונות מתוך מאגר MNIST

**MS-COCO-** הוא מערך נתונים בקנה מידה גדול ועשיר לזיהוי אובייקטים, פילוח וכתוביות. מכיל כ-330 אלף תמונות שמתוכם יותר מ200 אלף מתויגות, מכיל מספר רב של קטגוריות )כ-80) ובנוסף מכיל גם כ-91 תת קטגוריות.מערך נתונים זה עושה גם שימוש ברשת CNN ובהקשרים שמיוחסים לחלק מהתמונות.

T**he Street View House Numbers (SVHN)-** דומה מאוד לmnist – אבל מכיל יותר דאטא מתוייג וגם משתמש ברשת CNN על מנת לסווג את התמונות של מספרי בניינים ל10 קטגוריות.

**Open Images Dataset-** מאגר של כמעט 9 מיליון קישורים לתמונות, מאגר זה מכיל גם סט ואלידציה (כ-21260 תמונות) וגם סט מבחן (כ-125436). מה שמייחד כה את המאגר זה הוא כמות הקטגוריות שיש בו, שהיא מעל ל5000 קטגוריות שונות. מערך זה ידוע גם בשימוש שלו בresnet101- אשר נותן תוצאות טובות וגם ברשת CNN.

# **3. מבוא לautoencoders**

## **3.1 מהו autoencoder (מקודד אוטומטי)**

בשנים האחרונות, ראינו פריחה מהירה מאוד בתחום של הבינה המלאכותית, היא מיושמת במספר תחומים כמו בריאות, ביטוח, בנקאות וכו'. כל השימוש בבינה מאלכותית ורשתות נוירונים בפרט דורש הרבה מאוד משאבים מהמחשב , וכעת זה אפשרי כי יש לנו GPUs טובים ומנגנוני אחסון מהירים ויעילים יותר.

אבל אפשר לתהות מה הקשר של אחסון ועיבוד כאשר אנו מדברים על בינה מלאכותית ורשתות נוירונים. אז אחסון הוא מה שבמיוחד הופך את המקודדים האוטומטיים לבולטים בלמידת מכונה. מקודדים אוטומטיים הם מודלים בפיקוח עצמי (כפי שהסברנו על unsupervised) שיכולים ללמוד לדחוס את נתוני הקלט ביעילות. למרות שזה לא מקרה השימוש היחיד של מקודדים אוטומטיים, כלומר יש עוד הרבה.

אז מה בעצם autoencoders (מקודדים אוטומטיים) ? מקודדים אוטומטיים הם מודלים של למידת מכונה בפיקוח עצמי המשמשים להקטנת גודל נתוני הקלט על ידי יצירה מחדש שלהם. מודלים אלה מאומנים כמודלים של למידת מכונה בפיקוח (supervised) ובמהלך הסקת מסקנות, הם פועלים כמודלים ללא פיקוח (unsupervised) ולכן הם נקראים מודלים בפיקוח עצמי. מקודד אוטומטי, במטרתו, מפחית ממדי נתונים על ידי לימוד כיצד להתעלם מהרעש בנתונים. Autorncoder היא בעצם רשת נוירונים אשר יש לה מבנה שבה לפתור כמה מקרים , אך ראשית כדי להבין לעומק יותר את ה-autoencoders נסביר ממה הם מורכבים ומה המבנה שלהם.

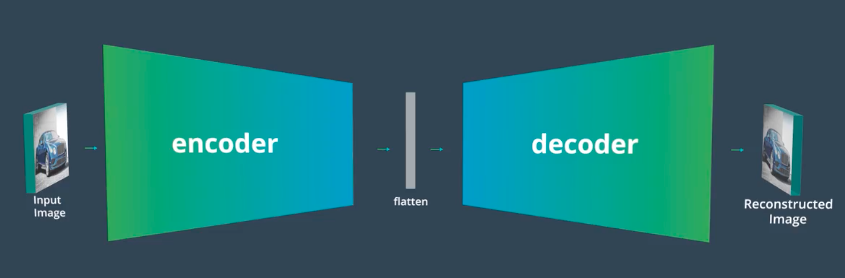
## **מבנה ה-autoencoder (encoder,decoder)**

מבנה הautoencoder מורכב מ4 חלקים עיקריים כפי שנסביר כעת וגם נראה באיור 12:

1. Encoder(מקודד) – בחלק זה המודל לומד כיצד לצמצם את ממדי הקלט ולדחוס את נתוני הקלט לייצוג מקודד.
2. Bottleneck- שהיא השכבה המכילה את הייצוג הדחוס של נתוני הקלט. זהו הממד הנמוך ביותר האפשרי של נתוני הקלט, לפי עיצוב הרשת שהדרנו.
3. Decoder (מפענח) – בחלק זה המודל לומד כיצד לשחזר את הנתונים מהייצוג המקודד כדי שיהיה קרוב ככל האפשר לקלט המקורי.
4. Reconstruction (אובדן שחזור)- בחלק זה אנו בעצם מודדים את ביצועי ה-decoder כלומר אנו בודקים כמה קרובה הפלט לקלט המקורי.

ולאחר השלב הרביעי האימון של הרשת כולל את שיטת ה- back propagation(עליה הסברנו במבוא) על מנת לצמצם את האובדן בחלק של השחזור של הרשת.

איור 12: 4 החלקים שמהם מורכב autoencoder



## **3.2 שימושים של autoencoders**

בחלק זה אנו נעבור על חלק מהשימושים של autoencoders נציג אותם ונסביר מה מטרתם.

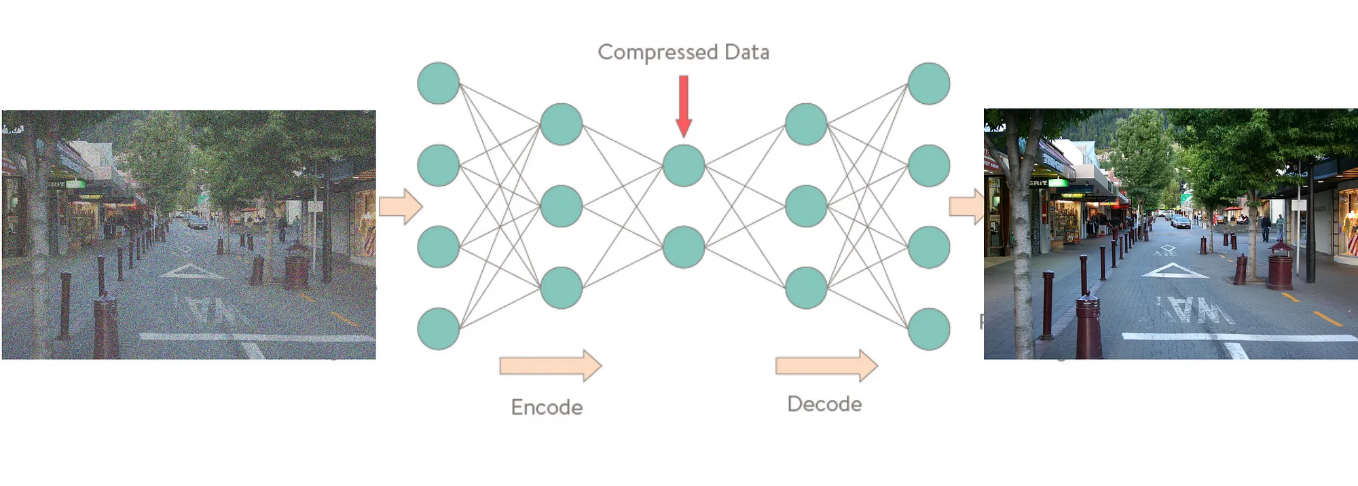
File Compression (דחיסת קבצים)- אחד מהשימושים העיקרים בautoencoders הוא שהם יכולים לצמצם את הממדיות של נתוני קלט, אשר אנו מתייחסים אליהם במשותף כדחיסה של קבצים. autoencoders עובדים עם כל מיני נתונים כמו תמונות, סרטונים ואודיו, זה עוזר בשיתוף וצפייה בנתונים מהר יותר ממה שיכולנו לעשות עם גודל הקובץ המקורי שלו.

Image transformation (טרנספורמציה של תמונה)- במקרה זה מקודדים אוטומטיים משמשים גם לשינויי תמונה, כלומר באמצעות דרך זו נוכל להפוך תמונות שחורות/לבניות לצבעוניות ולהיפך, כלומר אנו יכולים לדגום ולהוריד את נתוני הקלט ולהפך.

Image De-noising (ריטוש תמונה) – מקודדים אוטומטיים משמשים גם כטכניקות להסרת רעשים (הסרת רעשים של תמונה), מה שהופך אותו לבחירה הטובה ביותר עבור הורדת רעשים הוא שהוא אינו דורש שום אינטראקציה אנושית, לאחר שאומן על כל סוג של נתונים שהוא יכול לשחזר נתונים אלה עם פחות רעש מהתמונה המקורית.

בעבודה זו אנו נרחיב יותר לעומק על שימוש בautoencoders- עבור שימוש לריטוש תמונות (Image De-noising).

## **שימוש הautoencoder- לריטוש תמונות**

כפי שהסברנו autoencoder יכול לשמש אותנו לריטוש תמונות, כלומר ניתן לנקות תמונה מרעש. אבל איך בדיוק זה קורה. כפי שהסברנו autoencoder מורכב מ4 חלקים שבכל אחד מהם תמונה עוברת ובסופו של דבר מרטשת את עצמה כך שהיא תחזור ללא רעש, כפי שניתן לראות בתמונה 13.

איור 13 דוגמא לריטוש תמונה באמצעות autoencoder

אבל איך בדיוק כל חלק מבצע את הריטוש הזה, כלומר איך בסופו של דבר מתבצע תהליך הריטוש?

ראשית צריך להבין שautoencoders מרטשים את התמונות על ידי השחתת הנתונים בכוונה על ידי הפיכה אקראית של חלק מערכי הקלט לאפס. באופן כללי, אחוז צמתי הקלט המוגדרים לאפס הוא כ-50%, אך גם יש מקורות אחרים מציעים ספירה נמוכה יותר, כגון 30%. זה תלוי בכמות הנתונים וצמתי הקלט שיש לך.

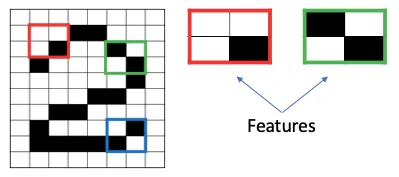
אבל איך בדיוק ה-autoencoder יודע איזה צמתים הם חשובים יותר ואיזה צמתים ניתן להפוך לאפס? . כדי לענות על זה אנו צריכים להבין ממה בנוי הencoder וה-decoder ואיך הם מחלצים את הנתונים מהתמונות

## **חילוץ נתונים ב-autoencoder לריטוש תמונות**

חילוץ הנתונים מתמונה, איך זה פועל? זה כולל את שלוש השכבות הבאות: שכבת הקונבולציה, שכבת reLu ושכבת ה-pooling.

גם פה תפקיד שכבת הקונבולוציה זהה לתפקיד ברשתות CNN נסביר בקצרה גם פה איך פועל ונשים לב לתוספת שיש על מנת לחלץ את הנתונים ואנו נראה ונבין מה יש גם בencoder וגם בdecoter.

**שכבת הקונבולוציה-** שלב הקונבולוציה יוצר חלקים קטנים רבים הנקראים featuars maps (בעיברית- מפות תכונה ) כפי שניתן לראות באיור 14.



איור 14 features map [25]

ריבועים אלו משמרים את הקשר בין פיקסלים בתמונת הקלט. בנוסף כל תכונה בתמונה נסרקת , תהליך זה נקרא filltring (סינון). כמו בתמונה 15

תמונה שמכילה קובייה, ריבוע, מלבן, עיצוב

התיאור נוצר באופן אוטומטי

Figure 5 איור 15 - תהליך ה-filttring[25]

לאחר הסריקה כל תכונה מייצרת תמונה מסוננת עם ציונים גבוהים וציונים נמוכים אם יש התאמה מושלמת, יש ציון גבוה בריבוע הזה, ואם יש התאמה נמוכה או שאין התאמה בכלל אז הציון יהיה נמוך או אפס בהתאמה.

עכשי נשים לב שככל שניקח יותר מסננים פירוש הדבר שחש יותר תכונות שהדגם יכול לחלץ. עם זאת, צריך לזכור שזה שיש יותר תכונות זה אומר זמן יותר זמן אימון.

בשכבת הקונבולוציה בנוסף יש את שלב ה-padding. בתהליך זה מתבצעת התאמה לתמונה כלומר , מתבצעת התאמה של אפס לתמונה המקורית או ויתור חלק מהתמונה המקורית וכל זה על מנת שנוכל לבצע את תהליך הfiltring- , כלומר אנו נוסיף שורות או נוותר על שורות כי לבצע את תהליך ההתאמה של מעבר ה-filttring .

בנוסף יש את ה-stide אשר קובע את את מספר הפקסלים הנעים עליהם, כלומר כאשר הstide הוא 1 אנו נעבור על פיקסל פיקסל כל פעם.

**מעבר ה-relu-** בחלק זה אנו עוברים ומוודאים ששכל הערכים הם אפס ומעלה על מנת שלא תהיה הפרעה בתהליך של הריטוש ולא תהיה הפרעה במתמטיקה שמתצעת.

**שכבת ה-polling-** בשלב זה אנו בעצם מבצעים הקטנה של התמונה , כלומר אנו לוקחים את הפיצרים הגבוהים ביותר וכך אנו בעצם מבצעים הקטנה ובעצם נוצרת סוג של תמונה חדשה מסוננת יותר. לאחר מכן אנו מפצלים את התמונות המסוננות הקטנות יותר ועורמים אותן לרשימה.

דבר שחשוב להשים לב אליו שעד כה תיארנו את התהליך לחילוץ הנתונים מהקלט שלנו ובעצם הקטנו את המימד שלו שזה בעצם התפקיד של הencoder .

אך בעצם המטרה שלנו זה לקחת תמונה עם רעש ולהחזיר אותה נקייה, וכאשר אנו מחלצים את הנתונים אנו נרצה אחר כך להחזיר אותם לגודל המקורי (לגודל הטבעי של התמונה) אז איך אנו מקבלים את התמונה בגודל המקורי שלה בחזרה ? פה יכנס התפקיד של הdecoder-.

**החזרת התמונה לגודל המקורי:**

תהליך זה גם משתמש בשכבת הקונבולוציה , שכבת ה-relu ושכבת ה-polling, אך השוני פה הוא בשכבת הקונבולוציה, בעצם בחלק זה (שה-decoder אחראי עליו) אנו מבצעים טרנספוז לשכבת הקונבולוציה (הפוך מקונבולוציה- נסביר על זה מייד), ותפקיד שכבת ה-relu וה-polling בעצם זהים לתפקיד שלהם בencoder ולכן השוני היחיד הוא בשכבת הקונבולוציה.

**טרנספוז קונבולוציה** :

זה בעצם הפוך מקונבולוציה רגילה , בחלק זה אנחנו יכולים להחזיר את הפחתת הגודל , ולהחזיר את התמונה לגודלה המקורי. פה בעצם לוקחים את הקלט שהחלק הראשון נתן לנו (encoder) ומגדלים אותו בפלט שלנו.

בדומה לשכבת הקונבולוציה , שכבת טרנספוז קונבולוציה עובר תהליך איטרטיבי של הכפלת ערכים במפת תכונות הקלט על ידי המסנן (filter) וחיבורם יחד ( כלומר מבצע מכפלה פנימית).

כך למעשה אנו יכולים להגדיל את התמונה לגדול המקורי שלה ולהחזיר את הקלט שלנו ללא הרעשים. (לפרט יותר.....)

אך כיצד הרשת מאמנת את עצמה ומחשבת את השגיאה מדימיון מהתמונה המקורית לתמונה שהחזרנו?

## **אימון ופונקציית שגיאה בautoencoder**

Feedforward ופונקציית שגיאה.

תוצאות על תמונות:

מחזיר תמונה קרובה ביותר לתמונה המקורית (ללא רעש?)...

# ביבליוגרפיה

1.What Is Deep Learning? 3 things you need to know – mathworks <https://www.mathworks.com/discovery/deep-learning.html>

2.Types of Neural Networks (and what each one does!) by Vansh Sethi. Toward Data Science Journal, 2019. - [Types of Neural Networks (and what each one does!) Explained | by Vansh Sethi | Towards Data Science](https://towardsdatascience.com/types-of-neural-network-and-what-each-one-does-explained-d9b4c0ed63a1)ז

3.Using Convolutional Neural Network for Image Classification, by Niklas Lang, towards data science Journal , Dec 5, 2021- <https://towardsdatascience.com/using-convolutional-neural-network-for-image-classification-5997bfd0ede4>

4. 8 Simple Techniques to Prevent Overfitting- by David Chuan-En Lin , towards data science Journal,2020 - <https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d>

5.Convolutional auto-encoder for image denoising of ultra-low-dose CT – sciencedirect , August 2017 - <https://www.sciencedirect.com/science/article/pii/S2405844016321600>

6.A Robust System for Noisy Image Classification Combining Denoising Autoencoder and Convolutional Neural Network by Imran Hossain , (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 1, 2018

7.A Better Autoencoder for Image: Convolutional Autoencoder , by Yifei Zhang , Australian National University ACT 2601, AU , 2018

8.Auto-encoder based dimensionality reduction – sciencedirect , [Volume 184](https://www.sciencedirect.com/journal/neurocomputing/vol/184/suppl/C), 5 April 2016, Pages 232-242 - <https://www.sciencedirect.com/science/article/abs/pii/S0925231215017671>

9.A noise robust convolutional neural network for image classification , Results in Engineering Volume 10, June 2021, 100225 - <https://www.sciencedirect.com/science/article/pii/S2590123021000268>

10.A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way, smith saha,2018 –

[A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way | by Sumit Saha | Towards Data Science](https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53)

11. Convolutional Neural Networks, Explained,   
Mayank Mishra-2020

<https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939>

12. Understanding Input Output shapes in Convolution Neural Network | Keras-

[Shiva Verma-2019-](https://shiva-verma.medium.com/?source=post_page-----f143923d56ca--------------------------------)

<https://towardsdatascience.com/understanding-input-and-output-shapes-in-convolution-network-keras-f143923d56ca>

13. The Softmax Function, Simplified- Hamza Mahmood,2018

<https://towardsdatascience.com/softmax-function-simplified-714068bf8156>

14. Activation Functions and Loss Functions for neural networks — How to pick the right one -Indraneel Dutta Baruah,2021

<https://medium.com/analytics-vidhya/activation-functions-and-loss-functions-for-neural-networks-how-to-pick-the-right-one-542e1dd523e0>

15. Understanding Loss Function in Deep Learning-Shankar297 — Published On June 20, 2022 and Last Modified On July 12th, 2022

<https://www.analyticsvidhya.com/blog/2022/06/understanding-loss-function-in-deep-learning/>

16. Beginners Ask “How Many Hidden Layers/Neurons to Use in Artificial Neural Networks- Ahmed Gad ,2018

<https://towardsdatascience.com/beginners-ask-how-many-hidden-layers-neurons-to-use-in-artificial-neural-networks-51466afa0d3e>

17. **6 basic things to know about Convolution-** [Madhushree Basavarajaiah](https://medium.com/@bdhuma?source=post_page-----daef5e1bc411--------------------------------), medium,2019

<https://medium.com/@bdhuma/6-basic-things-to-know-about-convolution-daef5e1bc411>

18. Understanding and Coding a ResNet in Keras, Priya Dwivedi, towardsdatascience,2019

<https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33>

19. Introduction to ResNets, Connor Shorten, towardsdatascience,2019

<https://towardsdatascience.com/introduction-to-resnets-c0a830a288a4>

20. Build an Image Classification Model using Convolutional Neural Networks in PyTorch, analyticsvidhya,2023

<https://www.analyticsvidhya.com/blog/2019/10/building-image-classification-models-cnn-pytorch/>

21. The first step towards Data Science, Deep Patel,2020

<https://deeppatel23.medium.com/first-step-towards-data-science-a4d8721e35f5>

22. Types of Machine Learning Algorithms You Should Know , Jose Fumo, towardsdatascience, 2017:

<https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861>

23. Neural Network: The Essence of Artificial Neural Network, Luthfi Ramadhan, towardsdatascience,2021:

<https://towardsdatascience.com/neural-network-the-essence-of-artificial-neural-network-c605eb32de56>

24. 25 Open Datasets for Deep Learning Every Data Scientist Must Work With, Pranav Dar,2021, analyticsvidhya:

<https://www.analyticsvidhya.com/blog/2018/03/comprehensive-collection-deep-learning-datasets/>

25. Transposed Convolutional Neural Networks — How to Increase the Resolution of Your Image, Saul Dobilas, towardsdatascience ,2022

<https://towardsdatascience.com/transposed-convolutional-neural-networks-how-to-increase-the-resolution-of-your-image-d1ec27700c6a>