הפקולטה להנדסה

המעבדה לעיבוד אותות

Audio-Visual Scene Analysis with Self-Supervised Multisensory Features

יוגב יוסף

נריה ליפשיץ

פרויקט שנה ד' לקראת תואר ראשון בהנדסה

מנחה: רננה אופוצ'ינסקי

מנחה אקדמי: פרופ' שרון גנות

אוקטובר 2022

תוכן עניינים

1 תקציר 3
2 תודות 4
3 מבוא 5
3.1 למידה עמוקה 5
3.1.1 מהי למידה עמוקה? 5
3.1.2 רשתות נוירונים 6
3.1.3 Semi Unsupervised Learning 10
3.1.4 CAM – Class Activation Map 11
3.2 הבעיה 12
3.2.1 הגדרת הבעיה 12
3.2.2 גישת הפתרון 12
4 בסיס הנתונים 13
4.1 הצגת בסיס הנתונים 13
4.2 עיבוד לפני האימון 14
5 המודל שלנו 16
5.1 ארכיטקטורה 16
5.2 היפר פרמטרים 20
5.2.1 Optimizers 20
5.2.2 Learing Rate 20
5.2.3 Weight Decay 21
6 תוצאות וסיכום 22
6.1 הצגת התוצאות 22
6.2 דיון וסיכום 26
7 ביבליוגרפיה 31

1 תקציר

אנו חווים את העולם שסביבנו דרך מגוון חושים ואמצעים: כאשר אנו משוחחים עם מישהו למשל, אנו שומעים את קולו דרך האוזניים ורואים את פיו ושפת גופו דרך העיניים. תיאום זה, מאפשר לנו להבין שישנו גורם משותף למראה ולקול, אשר יוצר את האותות הללו, ומאפשר לנו לחוות את האירוע בשלמותו דרך החושים שלנו.

בבואנו לפתור בעיה זו, של מיקום מקורות קול בסרטון, אנו רוצים לקבל השראה ולחקות את מערכת החושים האנושית, ולכן בחרנו בשיטה רב חושית, בה המודל מקבל משימה, אשר לצורך פתרונה נצפה שיידרש להתבונן בשילוב זה של אותות, במקרה זה – אות השמע והאות הוויזואלי.

כמו כן, על מנת להימנע מהצורך בשימוש במידע מתויג, נקטנו בשיטה של למידה לא מפוקחת חלקית (semi - unsupervised training). המשימה שהמודל שלנו קיבל, הייתה להבחין בין סרטונים אשר מסונכרנים עם אות השמע שלהם, לבין כאלו שלא, כאשר אנו מצפים, שהמודל יצטרך לשים לב לתיאום בין האותות, ובעצם יוכל "למקם" עבורנו את מקורות הקול בסרטון.

לתוצאות של למידה זו יכולים להיות יישומים נוספים עבור משימות נוספות הקשורות לתיאום של שמע ומראה, כמו זיהוי פעולות והפרדת דוברים בסרטונים.

2 תודות

נרצה להביע את מירב הערכתנו למנחה שלנו גב' רננה אופוצ'ינסקי על ההדרכה המלמדת לאורך כל פרויקט הגמר ועל המסירות גם מעבר לשעות המוגדרות לצרכי הדרכה.
בנוסף, נרצה להודות לפרופ' שרון גנות, ראש מחלקת עיבוד אותות וגם המנחה האקדמאי שלנו.

לבסוף נרצה להביע הערכה רבה למשפחותינו היקרות שתמכו בנו לאורך כל הדרך גם בזמנים הקשים.

3 מבוא

3.1 למידה עמוקה

3.1.1 מהי למידה עמוקה?

ראשית, נרצה להציג שלושה מושגים: בינה מלאכותית (Artificial Intelligence - AI), למידת מכונה (Machine Learning - ML) ולמידה עמוקה (Deep Learning - DL):

המושג "בינה מלאכותית" מתייחס לכלים חישוביים, אשר יכולים להחליף בינה אנושית בביצוע של משימות מסוימות. בפשטות, זהו כל חישוב, קטע קוד או אלגוריתם אשר יכול לחקות לפתח או להפגין התנהגות או מודעות אנושית.

"למידת מכונה", לעומת זאת, זהו תת תחום של בינה מלאכותית, אשר מתייחס לבניית יישומים, אשר יכולים ללמוד מבסיס נתונים, ולשפר את הדיוק שלהם ככל שהאימון מתקדם, מבלי שתוכנתו למשימה ספציפית זו. למידת מכונה מכוונת לניתוח של תבניות בדוגמאות האימון, ע"מ להצליח בקבלת החלטות ולמידה. למידת מכונה כרוכה באיסוף כמויות גדולות של מידע ע"מ להשתמש בו לאימון ושיפור הפרמטרים ומבנה המודל. ככל שייאסף יותר מידע, נוכל לפתח מודלים חדשים ומדויקים יותר.

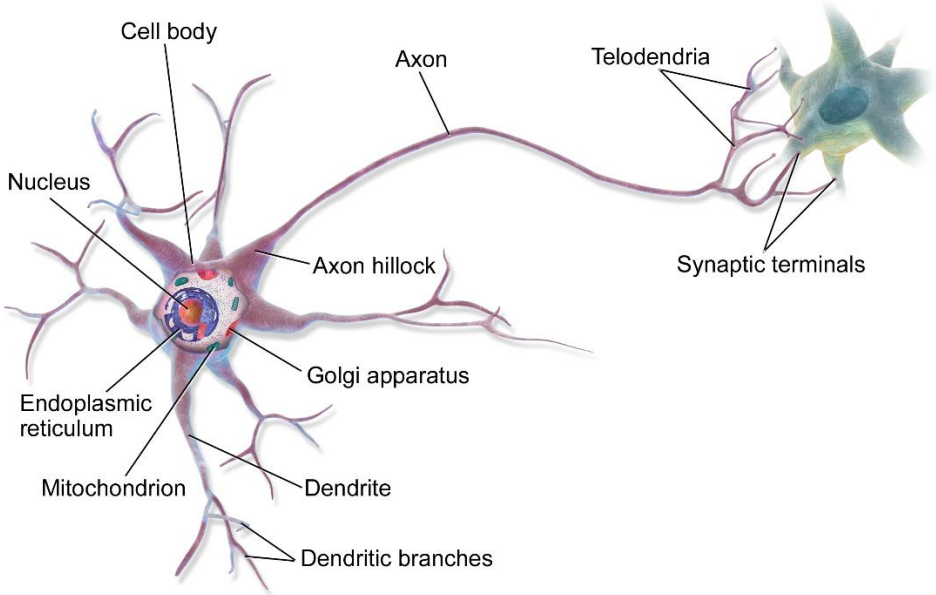
כעת, "למידה עמוקה" היא תת תחום של למידת מכונה, אשר מנסה לחקות את הדרך בה אנשים רוכשים סוגים מסוימים של מידע. באותה הדרך שהמוח האנושי סופג מידע דרך חמשת החושים, למידה עמוקה מעבדת את המידע הנכנס מערוצים שונים ומנתחת אותו, בעזרת מבנה אלגוריתמי שנקרא רשתות נוירונים מלאכותיות (Artificial Neural Network - ANN).

בפשטות, למידה עמוקה נבדלת משאר תחומי למידת מכונה באופן בו היא לומדת מהנתונים. כאשר לרוב למידת מכונה נדרשת למידע מתויג ומובנה, למידה עמוקה יכולה להתמודד גם עם דאטה שאינו מתויג ואינו מובנה. במקום להתסמך על תיוג בדאטה, ע"מ לזהות ולקטלג מידע, למידה עמוקה משתמשת ברשתות נוירונים לחלץ מאפיינים מהדאטה באופן שנעשה מוצלח יותר ויותר עם הזמן. תחומים אחרים בלמידת מכונה, מוצלחים ככל שיהיו, עדיין זקוקים להכוונה. למשל, אם מודל למידת מוכנה נותן חיזוי לא מדויק, המהנדס יצטרך להתערב ולבצע שינויים. במודל של למידת עמוקה לעומת זאת, המודל עצמו יכול לקבוע האם החיזוי שלו מדויק או לא בהסתמך על רשת הנוירונים שלו. עבור כל שכבה במודל, אלגוריתם הלמידה העמוקה ממצעת חישוב וחיזוי שוב ושוב, וכך משפרת את הדיוק ככל שהאימון מתקדם.

לסיכום, למידה עמוקה היא כלי מדעי ויישומי משמעותי, אשר כולל מודלים סטטיסטיים ופרדיקטביים, אשר מאפשרים לנתח כמויות עצומות של מידע.

3.1.2 רשתות נוירונים

רשתות נוירונים, הן מבנים אלגוריתמיים אשר קיבלו השראה מהביולוגיה של המוח האנושי. בקצרה, המוח מורכב ממיליארדי תאים הקרואים נוירונים, אשר קשורים בינם לבין עצמם ברשת קשרים מורכבת לאין שיעור. כאשר כל נוירון יכול להעביר אות אלקטרו-כימי, כתלות בשאלה כמה נוירונים הקשורים אליו העבירו לו אות כזה.



תרשים 1: מבנה סכמתי של נוירון. לנוירון מספר כניסות, המגיעות מנוירונים אחרים, דנדריטים. הנוירון יוציא אות דרך האקסון, כתלות במספר הנוירונים הקשורים אליו מהם יקבל אות.

באופן דומה, מבנה של רשת נוירונים מלאכותית בנוי משכבות שונות המכילות "נוירונים", כאשר כל שכבה קשורה לשכבה הקודמת, באופן דומה לנוירונים ביולוגיים. בכל שכבה מופעלים הנוירונים בתלות במוצא השכבה הקודמת, וכך מתקדם עיבוד המידע עד לתוצאה הרצויה.

באופן כללי, השימוש ברשתות נוירונים מתחלק לשלושה שלבים: א' - בחירת המודל. ב' – אימון המודל באמצעות הדאטה הקיים. ג' – יישום המודל המאומן על דאטה חדש.

באופן מעט יותר טכני, מודלים של רשתות נוירונים לומדים באופן הבא. מגדירים פונקציית מחיר (loss), אשר מחושבת בעזרת הדאטה של האימון, והפרמטרים הנלמדים של המערכת. פונקצייה זו מוגדרת באופן כזה שהערך שלה גבוה אם המודל לא מצליח לעשות חיזוי מוצלח, לעומת התיוג האמיתי של הדאטה.

לדוגמא, פונקציית המחיר בה אנו השתמשנו היא Binary cross entropy – BCE.

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

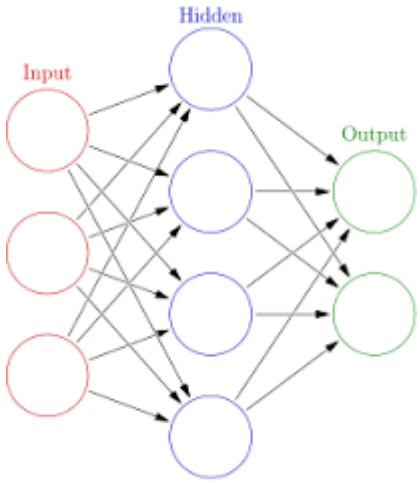
Binary Cross-Entropy / Log Loss

בעזרת שימוש בלוג, הפוקצייה מקבלת ערך מאוד גבוה עבור, עבור הסתברות לחיזוי מוטעה. בשלבים הבאים, אנו רוצים לשנות את הפרמטרים הנלמדים של המודל, באופן כזה שתוצאות פונקציית המחיר ימוזערו.

אנו עושים זאת בעזרת שיטה שנקראת :gradient descent בכל איטרציה, אנו מחשבים את הנגזרת של פונקציית המחיר ביחס לכל אחד מהפרמטרים הנלמדים, ומעדכנים את ערכו בכיוון זה, בתקווה שלבסוף נגיע למינימום גלובלי של המערכת, והמודל יצליח לתת פרדיוקציות כמה שיותר מדוייקות.

Fully connected NN

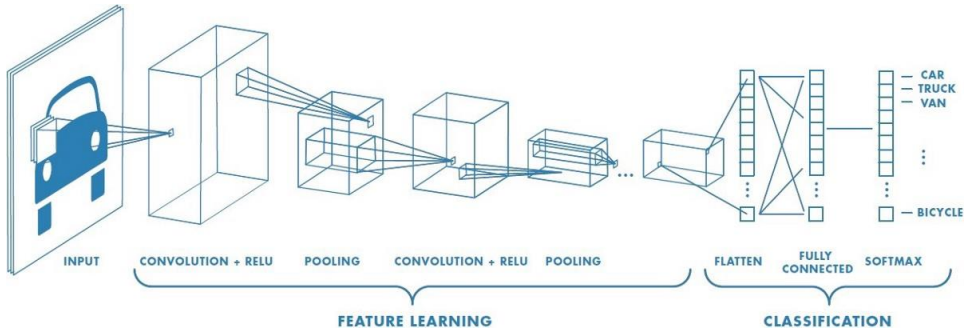
קיימים סוגים שונים של רשתות נוירונים, כאשר ניתן לשלב בין מספר ארכיטקטורות באותו מודל .נפרט על המבנים בהם השתמשנו בפרויקט זה. הסוג הבסיסי ביותר נקרא Fully connected NN, ממש כמו בתרשים לעיל בכל שכבה לכל נוירון מחוברים כל הנוירונים מהשכבה הקודמת.



תרשים 2: מבנה סכמתי של רשת נוירונים בעלת שכבה חבויה אחת. ערכו של כל נוירון נקבע לפי סכימה מסוימת של כל הנוירונים מהשכבה הקודמת המחוברים אליו.

CNN – Convolutional

מבנה נפוץ נוסף, בעיקר כאשר עוסקים במידע חזותי הוא CNN – Convolutional, זהו מבנה של רשת נוירונים אשר משמר את היחסים המרחביים בין המאפיינים, ולכן מוצלח הרבה יותר בעיבוד של מידע חזותי. להלן תרשים של CNN טיפוסי :



תרשים 3 : מבנה סכמתי של CNN. בכל שכבה מצמצמים את מימדי התמונה ע"י סדרה של פילטרים, ע"מ למצוא מאפיינים מרחביים אשר מסייעים להחלטת המודל בתהליך הלמידה.

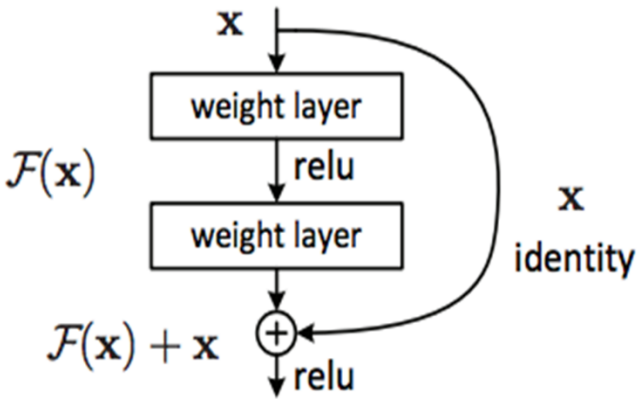
בארכיטקטורה זו, בכל שכבה אנו מבצעים קונבולוציה דו מימדית של המאפיינים, עם סדרה של פילטרים, שאת הפרמטרים שלהם המודל לומד ומעדכן על מנת למקסם את הביצועים. כך מימדי התמונה מצטמצמים, והמודל בעצם יכול להתייחס למאפיינים מרחביים הולכים וגדלים של הקלט.

ארכיטקטורה זו מומשה לראשונה בשנות ה-80, אולם היו לה יישומים מועטים בשל מחסור בכח חישוב ובמאגרי מידע גדולים מספיק. בשנת 2012 מומשה הרשת AlexNet, אשר השתמשה בCNN והגיע לתוצאות טובות בפער ניכר בתחום של סיווג תמונות, לעומת שיטות אחרות. כיום רשתות עמוקות מאוד המבוססות על CNN משיגות, בתחום של סיווג תמונות, תוצאות טובות יותר מאנשים העושים את אותה המשימה (!).

כיוון שהמודל שלנו עוסק בסרטונים, אנו משתמשים בקונבולוציות בשלושה מימדים, ע"מ לשמר מאפיינים במימד הזמן, ולא רק במימדים המרחביים.

Residual neural network - ResNet

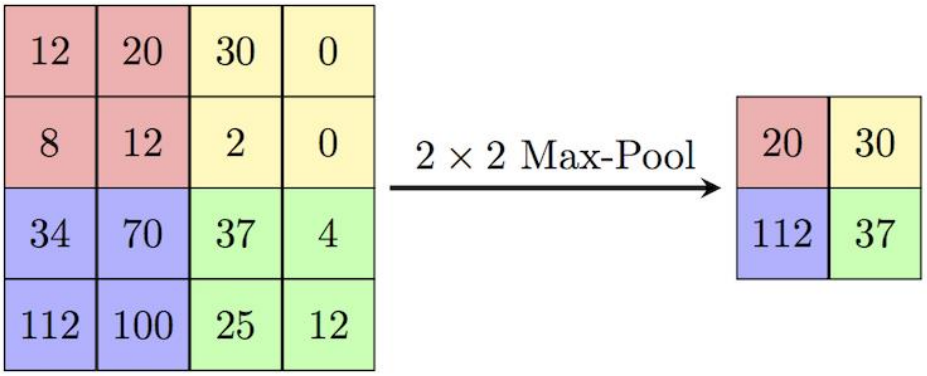
שכבת ResNet היא טכניקה שמומשה לראשונה בשנת 2015 במאמר Deep Residual Learning for Image Recognition. טכניקה זו עוזרת לנו להימנע מבעיית הגרדיאנטים המתאפסים שיש לה נטייה לקרות במודלים עם שכבות עמוקות ע"י זיכרון למרחק גדול יותר לאחור בעומק השכבות ברשת. בטכניקה זו אנו יוצרים מעקף בין שכבה קדומה יותר לשכבה מאוחרת יותר וכך מתקיים הזיכרון.



תרשים 4 : מבנה סכמתי של ResNet. במודל קיים מעקף כך שחלק מהמידע מדלג על כמה שכבות ללא שום למידה כלשהי ולאחר מכן עושים חיבור כלשהו עם המידע שנלמד.

Pooling layer

בשכבה זו, אנו לרוב מצמצמים את מימדי הקלט, באופן כזה שכל חלון בגודל שקובעים – מצטמצם לפיקסל אחד, לפי סוג הpooling. אנו למשל השתמשנו בmaxPooling, זאת אומרת, שכל חלון מוגדר מראש, מצטמצם לפיקסל בודד, אשר מקבל את הערך של הפיקסל המקסימלי מהחלון.

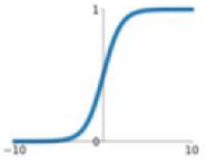
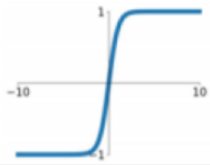
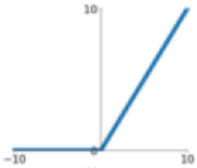
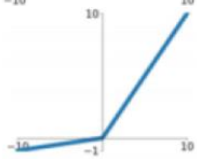
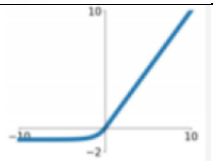


תרשים 5 : מבנה סכמתי של Max-Pooling. אנו מצמצמים את המידע לפי גודל של 2X2 כך שהמספר הגדול ביותר נבחר.

פונקציות אקטיבציה

פונקציות אקטיבציה הן פונקציות חשובות שאנחנו שמים במוצא של כל שכבת נוירונים. אחד המאפיינים העיקרים של פונקציות האקטיבציה זה שהן לא לינאריות מהסיבה הפשוטה, כל הרשת זה בעצם הרכבה של פונקציות אחת על השנייה והרכבה של 2 פונקציות לינאריות תיתן לנו פונקציה לינארית. לכן על מנת לפתור בעיות קצת יותר מורכבות ולא לינאריות אנו חייבים להכניס לוגיקה שהיא לא לינארית, הדבר נעשה ע"י פונקציות האקטיבציה.

דוגמאות לפונקציות אקטיבציה:

Sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$ 	Sigmoid
tanh $\tanh(x)$ 	Tanh
ReLU $\max(0, x)$ 	ReLU
Leaky ReLU $\max(0.1x, x)$ 	Leaky ReLU
ELU $\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$ 	ELU

Semi Unsupervised Learning 3.1.3

כפי שהזכרנו, בהרבה מהמקרים, למידה עמוקה דורשת כמויות עצומות של מידע מתויג. ואכן, אחת הבעיות העומדות בפני חוקרים ומהנדסים של למידה עמוקה הוא השגת מידע איכותי ומתוייג לצורך אימון המודלים, וחברות העוסקות בתחום משקיעות כסף ומשאבים רבים במהלך הרכישה של מידע שכזה.

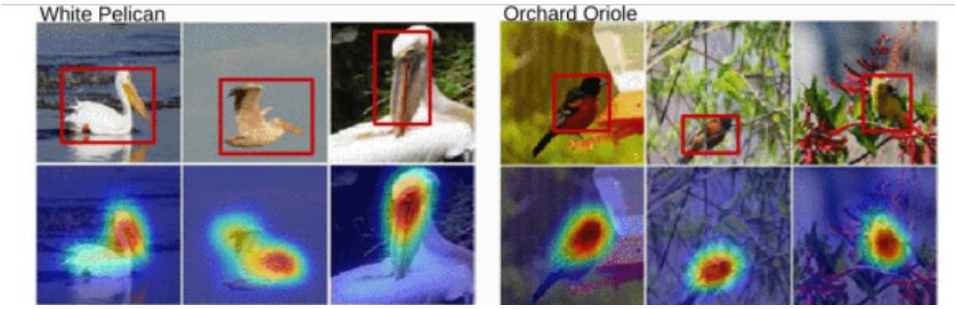
אומנם, ישנן שיטות אימון, ואלגוריתמים של למידת מכונה שלא דורשים כלל דאטה מתויג. אולם המודל אותו אנו מאמנים אכן דורש מידע מתויג.

על מנת להצליח לאמן את המודל שלנו מבלי להצטרך לשלם הון על מידע מתויג, השתמשנו ב"למידה בלתי מונחית חלקית" – הווי אומר, אנו יצרנו בעצמנו את המידע המתויג. כאמור, המשימה הראשונית של המודל שלנו היא להבחין בין סרטונים המסונכרנים עם אות השמע שלהם לבין כאלו שלא. במהלך הכנת הדאטה לאימון לקחנו כ50% מהסרטונים, וע"י סקריפט מקומי הוצאנו את השמע מסנכרון, ותייגנו אותם כלא מסונכרנים. כך השגנו מידע מתויג, מבלי להשקיע כסף ומאמצים לשם כך.

3.1.4 CAM – Class Activation Map

CAM הינה שיטה לבחינת "אזורי העניין" של המודל. מאמר מפורסם שיצא לאור ב2016, מציג איך ניתן לראות את האזורים בהן מודל הCNN התמקד ע"מ לקבל החלטה.

לדוגמא, להלן פלט שהתקבל מיישום CAM על מודל של סיווג תמונות:



תרשים 6 : הדגמה של יישום CAM על רשתות של זיהוי תמונות.

למעשה, במקרה שלנו, הפקנו את הCAM באופן הבא:

כאשר ההסתברות לקבל החלטה נתונה ע"י המשוואה הבאה:

$$p(y \mid I_x, A_x) = \sigma(w^\top f(I_x, A_x))$$

כאשר y הוא ההחלטה הבינארית, σ מייצגת את פונקציית הסיגמואיד, ו- f היא המוצא של שכבת הקונבולוציה האחרונה.

הביטוי הבא:

$$|w^\top f(I_x, A_x)|$$

ייתן לנו את המשקלים של הCAM. אולם, כיוון שגודל הקלט מצטמצם מאוד עד שלב זה, על מנת שנוכל להציג את הCAM ע"ג הווידאו המקורי, הצטרכנו "למתוח" את הCAM ע"י אינטרפולציה.

תוצאות ודוגמאות של שלב זה יוצגו בחלק ג' – תוצאות ודיון.

3.2 הבעיה

3.2.1 הגדרת הבעיה

אחת מהבעיות הקשורות לניתוח וידאו, היא מיקום (Localization) של מקורות קול בסרטון. לפתרון בעיה זו באופן מוצלח יכולים להיות יישומים רבים. למשל, זיהוי של מפגעים או אנשים במצוקה מתוך צילומים של מצלמות אבטחה.

וכן, בשילוב של מתודות נוספות לניתוח של סרטונים ותמונות, כמו סגמנטציה והפרדת דוברים, ניתן יהיה לערוך בקלות סרטונים ולהסיר מהם דמויות, שזוהו כמקור הסאונד ע"י הסגמנטציה, ואת פס הקול שלהן, שהופרד ע"י הפרדת דוברים. האפשרויות הן בלתי מוגבלות.

יישום חשוב נוסף שיכול להיות, הוא ויזואליזציה מינימלית של דוברים ומקורות קול משמעותיים בסרטים וסרטונים, באופן כזה שיאפשר לכבדי שמיעה להבין טוב יותר את הסצנה המוצגת על המסך.

3.2.2 גישת הפתרון

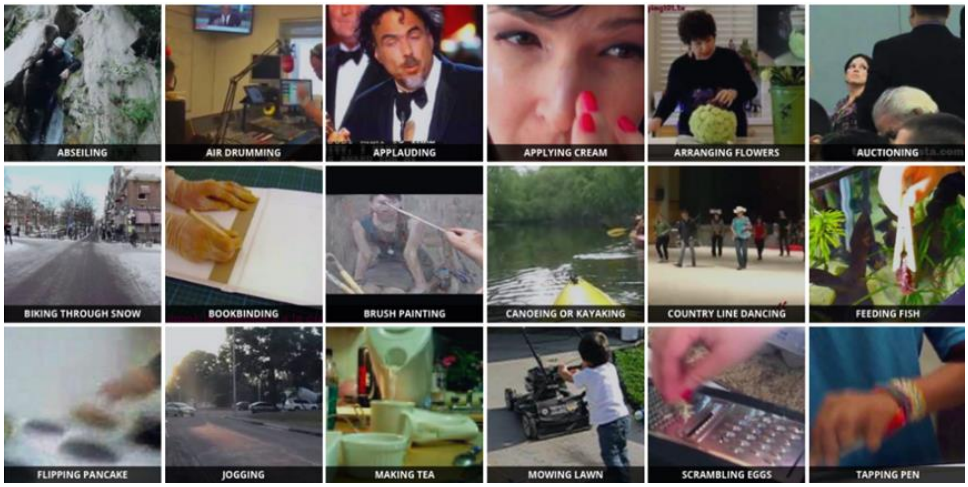
כאמור, האופן בו אנו ניגשים לפתרון הבעיה היא מודל רב-חושי (Multisensory). ראשית, חיפשנו משימה שעל מנת לבצע אותה, המודל יצטרך "לשים לב" למקורות הקול בסרטון. המשימה שנתנו היתה אבחנה בין קליפ עם אודיו מסונכרן, לבין כזה שאינו, מתוך ציפייה, שבדומה לאדם שמנסה להחליט במשימה דומה, המודל ייאלץ לתת משקל גבוה בהחלטתו למקורות הסאונד ע"מ לעשות החלטה טובה. לאחר מכן, כפי שהוסבר לגבי הCAM, ניתן יהיה לקחת חלק משכבות המודל, ולהתבונן על החלקים להם המודל נתן משקל גבוה בהחלטה.

4 בסיס הנתונים

4.1 הצגת בסיס הנתונים

לאחר מחשבה על אופי בסיס הנתונים שאותו נצטרך על מנת שיעזור לנו למקסם את הלמידה וזיהוי מיקום מקורות הקול בסרטונים בחרנו בבסיס הנתונים "Kinetics" . בסיס נתונים זה פורסם לראשונה במאי 2017 ע"י חברת "DeepMind" והוא קיבל את האישור והרישיון של גוגל היות והוא מכיל סרטונים מYouTube. בסיס נתונים זה מחולק ל3 קטגוריות: "Kinetics-400", "Kinetics-600" "Kinetics-700", כאשר "Kinetics-700" עבר עדכון קל ב2020 ולכן יש גרסה מחודשת שלו של "Kinetics-700-2020". אנחנו בחרנו להשתמש ב"Kinetics-700-2020" היות ויש שם מגוון רחב יותר של פעולות – כל מספר מייצג את מספר הפעולות האנושיות שיש בו.

בסיס נתונים זה מכיל כ650 אלף סרטונים ומכסה כפי שאמרנו 700 פעולות אנושיות. הסרטונים בבסיס נתונים זה כוללים פעולות כמו נגינה בכלי נגינה וכן אינטראקציה בין אנשים כמו חיבוק, לחיצות ידיים או מריבה. בכל אחת מהפעולות יש מעל 700 קטעי וידאו שכל אחד מהם נמשך כ10 שניות.



תרשים 7: פריימים לדוגמה מתוך סרטונים שקיימים במסד הנתונים.

נזכיר כי המשימה העיקרית שלנו היא זיהוי מקורות רעש ולכן בסיס נתונים זה הוא מצוין בשבילנו מכמה סיבות:

- יש חלוקה לקטגוריות כך שיכלנו להתמקד בפעולות קצת יותר אינפורמטיביות בחלק מהזמן כמו מחיאות כפים, שיח של אדם או לתת מכה עם פטיש.
- בסיס נתונים זה מכסה פעולות אנושיות כך שרוב הסרטונים מוקלטים ע"י אנשים ולכן לרובם אין רעשי רקע שיכולים להיות בהרבה סרטונים אחרים כמו רוח חזקה/גשם.

4.2 עיבוד לפני האימון

אחד השלבים המרכזיים בלמידה מכונה הוא טעינת והכנת הdata, לשלב זה חשיבות רבה על איכות הלמידה והתוצאות שלה. פרמטר נוסף וחשוב הוא הזמן שאותו לוקח לטעון batch כלשהו בזמן האימון של הרשת, כאשר יש פה איזון שצריך לשמר. מצד אחד יש רצון לאפשר גמישות רבה על מנת שנוכל לשנות את המודל או את האוגמנטציות אשר נעשות על הdata מצד שני אם יהיו יותר מידי פעולות שמתבצעות על הdata בזמן האימון של הרשת הדבר יאריך את זמן האימון. לכן את שלבי עיבוד הdata לפני האימון ביצענו בהדרגה רבה על מנת לאפשר גמישות מקסימלית לשינויים במודל האימון.

הכנו סקריפט CuttingVideosScript.py שבו עשינו מודיפיקציה לdata על מנת שיהיה יותר מותאם לקליטה לפני אימון וכדי שנעשה חלק מעיבוד הdata עוד לפני זמן האימון שצפינו שיהיה ארוך מאוד. בסקריפט זה צמצמנו את כל הסרטונים לפי פרמטרים מסוימים שאותם קבענו:

- קצב דגימת הווידאו – 30 פריימים לשנייה
- קצת דגימת האודיו – 44100 דגימות לשנייה
- אורך סרטון – הגדרנו שכל סרטון יהיה באורך של 4 שניות

בנוסף, על מנת לממש את המשימה שלנו ע"י שימוש ב random.rand() עשינו cyclic shift ל50% מהסרטונים כאשר כל שם של סרטון הוא מספר ואם הסרטון אינו מסונכרן אז הוספנו בסוף השם שלו "s_" כך ידענו להבדיל לצרכי דיבאג.

כל מה שנותר הוא לשמור את הסרטונים כקבצי mp4 חדשים ולהשתמש בטבלת גישה אליהם שאותה שמרנו כקובץ csv כאשר עמודה ראשונה הייתה הנתיב לסרטון החדש ועמודה שניה זה הlabel האם האודיו והווידאו מסונכרנים או לא.

לאחר מספר ניסיונות אימון הבנו כי עדיין יש צוואר בקבוק בזמן טעינת הdata ולכן החלטנו להחמיר במודיפיקציות שאותן ביצענו ואף באופי שמירת הקבצים. ניסינו לשמור קובץ pth שאותו torch יודע לטעון עם API מסודר שלו ועשינו השוואה בין זמן טעינת הווידאו אל מול זמן טעינת הpth ע"י טעינת 50 סרטונים ב2 התצורות:

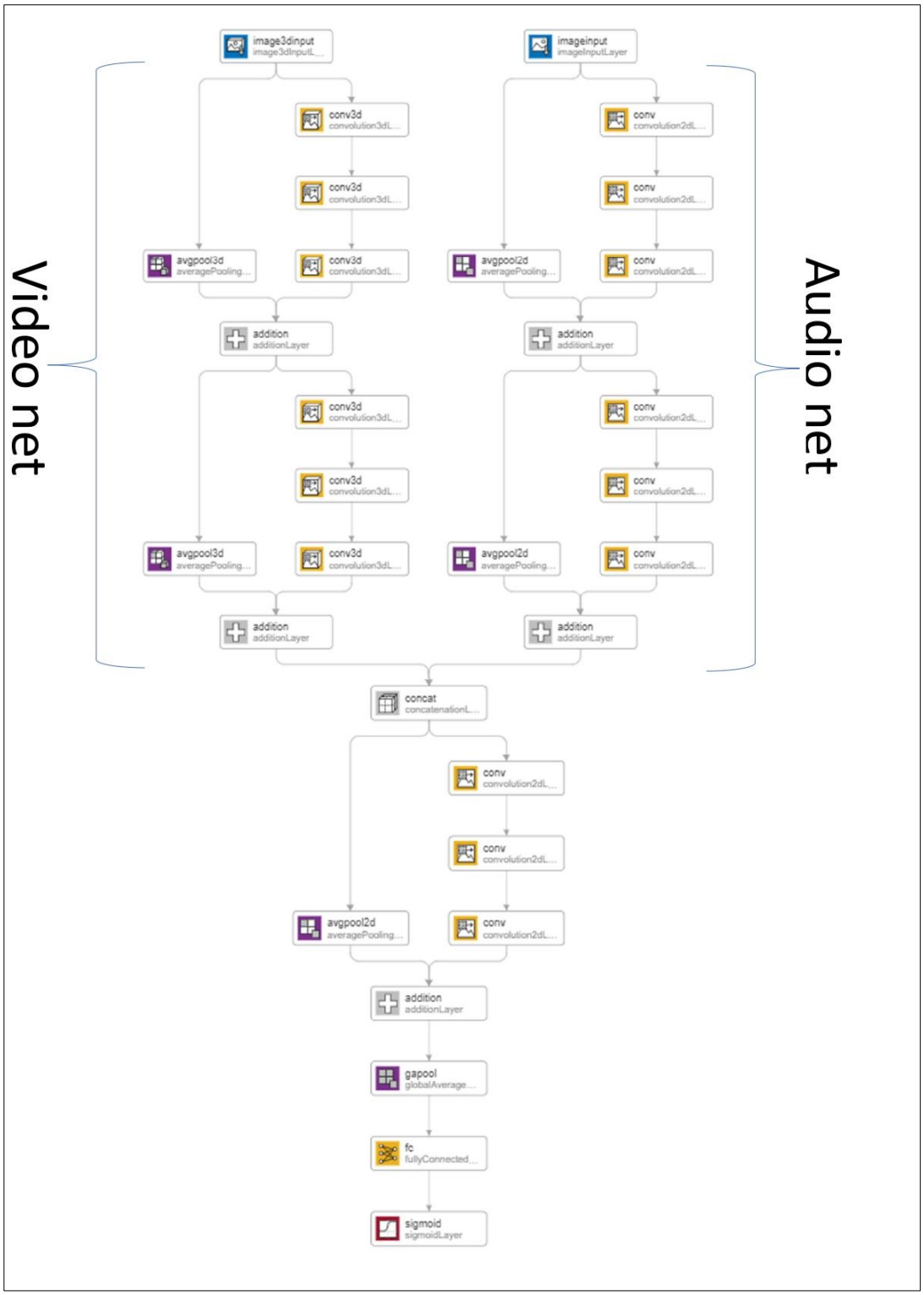
```
read time =  16.522755384445 19
 load time =  2.2666237354278564
```

כפי שניתן לראות זמן הטעינה של קבצי הpth מהיר פי 8, כמובן שהדבר לא בא בחינם והמחיר אותו שילמנו הוא בגודל הקובץ. בממוצע קובץ pth שקל פי 10 לעומת קובץ מקביל שלו בצורת mp4 היות ומדובר בfloat לעומת int שנשמרים בmp4.
לבסוף החלטנו גם לעשות את האוגמנטציות שתכננו לעשות בזמן טעינת הdata כדי לאפשר את זמן הריצה הטוב ביותר היות והמודל מאוד התקשה לעמוד בקצב למידה מספק.

דבר נוסף, על מנת לאפשר גמישות בtests יצרנו טבלאות גישה בקובץ csv עבור כל אחת מהפעולות שקיים בבסיס הנתונים, דבר זה אפשר לנו לעשות tests עבור פעולה ספציפית ולאו דווקא בצורה רנדומלית. הדבר חשוב מכיוון שיש פעולות מסוימות שיכולות להיות מאוד פשוטות לזיהוי מקורות הרעש כמו הכאה בפטיש לעומת פעולות מסוימות שמאוד קשות אפילו לעין האנושית כמו ריקודים כפי שנרחיב בהמשך.

5 המודל שלנו

5.1 ארכיטקטורה



תרשים 8 : תרשים סכמתי של מבנה המודל שבחרנו – יצרנו באמצעות תוסף לMATLAB.

המודל שלנו מורכב מהשכבות השונות עליהן פירטנו בחלק התיאורטי.
בשלב זה נפרט באופן טכני יותר על חלקי המודל, ומימדי הדאטה בכל שלב בו. המודל מתחלק לשלושה חלקים. החלק המוזן באות האודיו (Audio net), חלק המוזן באות הווידאו (Video net), ולאחר מכן, שני המוצאים מחוברים יחד, ונכנסים לחלק השלישי (Fused net), בו הם עוברים עד להחלטה הבינארית – האם הסרטון מסונכרן או לא.

החלק הראשון – כאמור, בחרנו להשתמש בסרטונים באורך של 4 שניות, כאשר קצב דגימת האודיו היא 44.1 KHz, ז"א שאות השמע שלנו מכיל 176,400 דגימות. כמו כן, כפי שיוסבר בחלק הדיון, לפני הכניסה למודל, איפסנו את 5000 דגימות השמע הראשונות.
כיוון שקצב הדגימה של אות השמע גדול בהרבה מקצב הדגימה של אות הווידאו, הצטרכנו להוריד משמעותית את ממדי האות. עשינו זאת באמצעות סדרה של שכבות קונבולוציה חד ממדיות, כאשר אחרי כל שכבת קונבולוציה הוספנו שכבת אקטיבציה לא לינארית, וכן הוספנו שכבות נרמול, ושכבות קיצור. להלן טבלה, המסכמת את השכבות והגדלים של החלק הראשון:

שם השכבה	סוג	Kernel	הערה	ממדים בכניסה	ממדים ביציאה
גודל הקלט ההתחלתי, פיצול RES					(1, 176400)
audConv1	conv1D	65	Stride = 4	(1, 176400)	(64, 44084)
audPool1	maxPool1D	4	Stride = 4	(64, 44084)	(64, 11021)
elu + audBatch1	BatchNorm1d			(64, 11021)	(64, 11021)
audConv2	conv1D	1	Stride = 4	(64, 11021)	(128, 2756)
audBatch2+מיזוג RES, פיצול RES	BatchNorm1d			(128, 2756)	(128, 2756)
audConv3	conv1D	15	Stride = 4	(128, 2756)	(128, 686)
elu+audConv4	conv1D	16	padding = 9	(128, 686)	(128, 689)
audBatch3, מיזוג RES, פיצול RES	BatchNorm1d			(128, 689)	(128, 689)
audConv5	conv1D		Stride = 4 padding = 6	(128, 689)	(128, 172)
audPool2	maxPool1D	4	Stride = 4	(128, 172)	(128, 28)
audConv6	conv1D	5		(128, 28)	(128, 24)
audBatch5	BatchNorm1d			(128, 24)	(128, 24)

החלק השני - כאמור, חלק זה עוסק בעיבוד אות הווידאו. הקלט שהוכנס למודל הוא סרטונים באורך 4 שניות, בקצב של 30 fps, דהיינו, 120 פריימים. כמו כן, שינינו את הממדים של כל פריים ל-224X224 פיקסלים. כמו כן, כל פיקסל מכיל שלושה ערוצים של פורמט RGB, ולכן הממדים של כל סרטון הם - (120, 224,224,3), אבל בכניסה למודל, החלפנו את הממד של הערוצים להיות הראשון, ולכן ממדי הכניסה של הקלט הם: (3,120, 224,224).

כמו בחלק הקודם, להלן טבלה המפרטת את סדר השכבות והגדלים:

שם השכבה	סוג	Kernel	הערה	ממדים בכניסה	ממדים ביציאה
גודל הקלט ההתחלתי, פיצול RES					(3, 120, 224, 224)
vidConv1	conv3d	(5, 7, 7)	stride=2	(3, 120, 224, 224)	(64, 58, 109, 109)
vidBatch1 + elu	BatchNorm3d			(64, 58, 109, 109)	(64, 58, 109, 109)
vidPool1	MaxPool3d	(1, 2, 2)	(1, 2, 2)	(64, 58, 109, 109)	(64, 58, 54, 54)
vidConv2	conv3d	3	stride=2	(64, 58, 54, 54)	(64, 28, 26, 26)
vidConv3	conv3d	3		(64, 28, 26, 26)	(64, 26, 24, 24)
vidConv4	conv3d	3		(64, 26, 24, 24)	(64, 24, 22, 22)
vidConv5	conv3d	(1, 3, 3)		(64, 24, 22, 22)	(64, 24, 20, 20)
vidBatch2 + elu +מיזוג	BatchNorm3d			(64, 24, 20, 20)	(64, 24, 20, 20)

החלק השלישי - בחלק זה הרחבנו את ממדי המוצא של חלק האודיו מ(128, 24) ל – (128, 24, 20, 20) בעזרת פקודת expand, ו-concat, כך שעבור 128 ערוצים מאות האודיו, ו-64 ערוצים מאות האודיו קיבלנו 192 ערוצים.

לסיכום ממדי הקלט הנכנס לחלק השלישי והאחרון של המודל הם (192, 24, 20, 20)/

גם כאן נציג טבלה מפורטת עבור השכבות והגדלים:

שם השכבה	סוג	Kernel	הערה	ממדים בכניסה	ממדים ביציאה
גודל הקלט ההתחלתי					(192, 24, 20, 20)
fusedConv1 פיצול RES	conv3d	1		(192, 24, 20, 20)	(128, 24, 20, 20)
fusedConv2	conv3d	3		(128, 24, 20, 20)	(128, 22, 18, 18)
fusedConv3	conv3d	3		(128, 22, 18, 18)	(128, 20, 16, 16)
fusedConv4	conv3d	3		(128, 20, 16, 16)	(256, 18, 14, 14)
fusedConv5	conv3d	3		(256, 18, 14, 14)	(512, 16, 12, 12)
fusedConv6	conv3d	(5,3,3)		(512, 16, 12, 12)	(128, 12, 10, 10)
fusedBatch1	BatchNorm3d			(128, 12, 10, 10)	(128, 12, 10, 10)
fusedBatch2+ מיזוג + פיצול	BatchNorm3d			(128, 12, 10, 10)	(128, 12, 10, 10)
fusedConv7	conv3d	3		(128, 12, 10, 10)	(256, 10, 8, 8)
fusedConv8	conv3d	3		(256, 10, 8, 8)	(512, 8, 6, 6)
fusedConv9	conv3d	(3, 2, 2)		(512, 8, 6, 6)	(128, 6, 5, 5)
fusedBatch3 + elu +מיזוג	BatchNorm3d			(128, 6, 5, 5)	(128, 6, 5, 5)
globalAvgPool	AvgPool3d	(6, 5, 5)		(128, 6, 5, 5)	(128, 1, 1, 1)
lastFC	linear			128	1

לצורך הפקת הCAM, חילצנו מהשלב השלישי גם את שכבת הקונבולוציה האחרונה, כאשר הממדים שלה הם (128, 6, 5, 5) וכן את המשקלים של השכבה הלינארית האחרונה.

5.2 היפר פרמטרים

5.2.1 Optimizers

בלמידה עמוקה יש את נושא הloss שמתאר לנו איך המודל שלנו מתפקד בנקודת epoch מסוימת ולפי השינוי בloss ניתן לדעת האם ובאיזה איכות המודל שלנו לומד. באופן עקרוני אנחנו צריכים למזער את הloss ככל שניתן כדי שהמודל יתפקד בצורה הטובה ביותר, כלומר לעשות אופטימיזציה וכאן נכנסים הoptimizers.

המטרה של האלגוריתמים של הoptimizers זה לחשב איך לשנות את משקלי המודל על מנת למזער את הloss. זה בלתי אפשרי לדעת בהתחלה מה המשקלים הטובים ביותר לאותו מודל, אבל ע"י ניסוי וטעייה שמבוססים על הloss הoptimizers יודעים לומר לאיזה כיוון הכי כדאי ללכת על מנת למזער את הloss.

יש מגוון רחב של optimizers ומתוכם בחרנו לחקור במודל שלנו את הבאים:

- SGD
- SGD with momentum
- Adam
- AdamW

לאחר מספר הרצות ובדיקות, החלטנו לבחור להשתמש ב SGD with momentum שהפיק את התוצאות הטובות ביותר.

5.2.2 Learning rate

פרמטר זה מגדיר כמה אנו מעדכנים את הפרמטרים של המודל בכל איטרציה. עדכון הפרמטרים נעשה לפי הנוסחה הבאה:

$$w_i' = w_i - \gamma \frac{\delta L}{\delta w_i}$$

כאשר γ מציין את הlr, והוא קובע את השיעור בו אנו מעדכנים את הפרמטרים. ישנה חשיבות רבה לקביעת ערך טוב לפרמטר זה, שכן אם נקבע ערך גבוה מדי, נוכל 'לדלג' בטעות על נקודות המינימום של פונקציית הloss. לעומת זאת, אם נקבע ערך קטן מדי, ייתכן שהלמידה תהיה מאוד מאוד איטית, כיוון שייקח הרבה איטרציות כדי להתכנס לנקודות המינימום.

לאחר ניסוי וטעייה, ראינו כי הערך הנותן את התוצאות המיטביות הוא $\gamma = 0.01$.

Weight decay 5.2.3

Weight decay הוא אלמנט בפונקציית הloss, אשר מטרתו היא להטיל רגולציה על גודל הפרמטרים הנלמדים, וכך למנוע overfitting.

כאשר מוסיפים את אלמנט הwd, פונקציית ה-loss תיראה כך:

$$L_{new}(w) = L_{original}(w) + \lambda w^T w$$

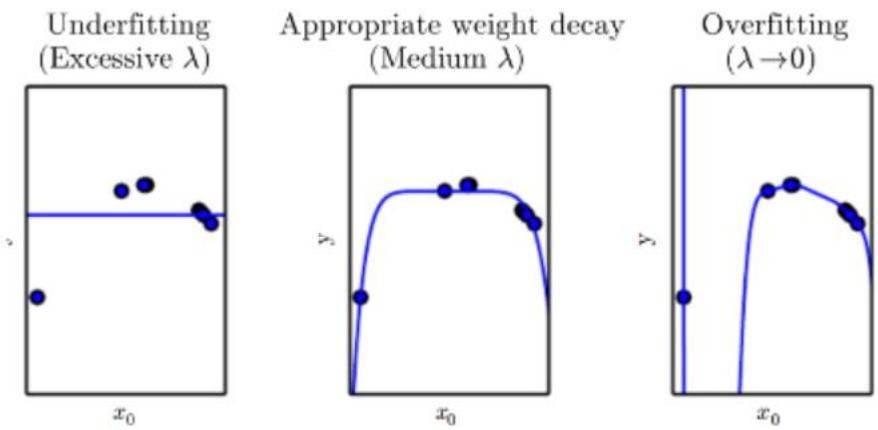
ז"א, אלמנט ה-λ, מוסיף ערך לפונקציית המחיר, אשר תלוי בגודל של הפרמטרים W, במקרה זה, ניתן לרשום:

$$L_{new} = L_{oroiginal} + \frac{\lambda}{\mathrm{n}} \sum w_i^2$$

וכך, כאשר מחשבים את הגרדיאנט עבור כל פרמטר, אלמנט הwd משפיע באופן הבא:

$$\frac{dL_{new}}{dw_i} = \frac{dL_{original}}{dw_i} + \lambda w_i$$

ההשפעה של אלמנט זה יכולה להיות משמעותית מאוד, כפי שמוצג בתרשים הבא:

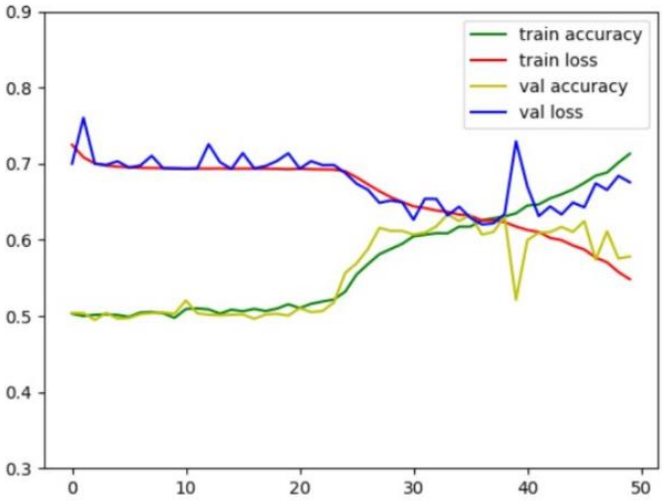


בחירה של ערך wd נכון, היא קריטית לתהליך למידה מוצלח. לאחר ניסוי וטעייה, כפי שנתאר בחלק הדיון, אנו בחרנו בערך $\lambda = 0.0001$.

6 תוצאות וסיכום

6.1 הצגת התוצאות

בסופו של דבר, לאחר ניסוי וטעייה רבים, עליהם נפרט בחלק הבא, המודל שלנו הפיק את גרף הלמידה הבא:



כפי שניתן לראות, בתהליך הלמידה ישנו שלב, בסביבות איטרציה 25, בה לפתע הלמידה הופכת להיות הרבה יותר טובה, והחיזוי נעשה מאוד מדויק. כפי שנסביר בהמשך, עבור המודל שלנו בחרנו דווקא את המודל שהתקבל לפני עלייה זו, באיטרציה 23.

ניתן לראות שהדיוק, גם עבור האימון וגם עבור הוולידציה נע סביבות ה55%. זו אולי לא נראית תוצאה מאוד גבוהה, אולם חשוב לזכור, שזוהי משימה שגם עבור בני אנוש היא לא קלה. למעשה, בבדיקה שערכו, מסתבר שאנשים מצליחים במשימה זו כ66%. כעת נתבונן, האם כמו שציפינו, המודל מתמקד במקורות הסאונד ע"מ לבצע החלטה.

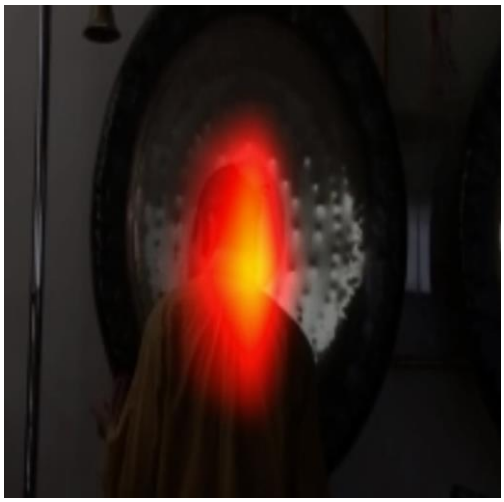
להלן 5 דוגמאות, לפריימים שנלקחו מתוך הסרטונים, בהם המודל הצליח במידה יפה למקם את מקורות הקול:

1. אישה מדברת בסרטון:

2. טרקטור עובד בשטח:

3. איש מתופף:

4. ניגון על גונג:



5. טרקטורון נוסע בשטח:

לעומת זאת, ישנם סרטונים בהם הוא היה פחות מוצלח:

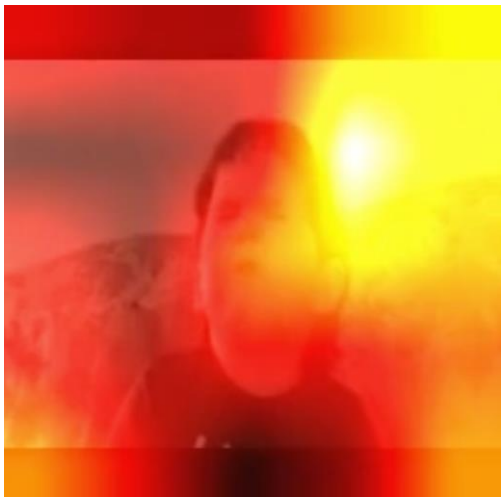
1. בנאדם רוקד במשרד:

2. תמונה מאיצטדיון:

3. עוד תמונה מאיצטדיון:

4. אנשים משקים עצים:

5. ילד עם מנגינה חזקה ערוכה ברקע:

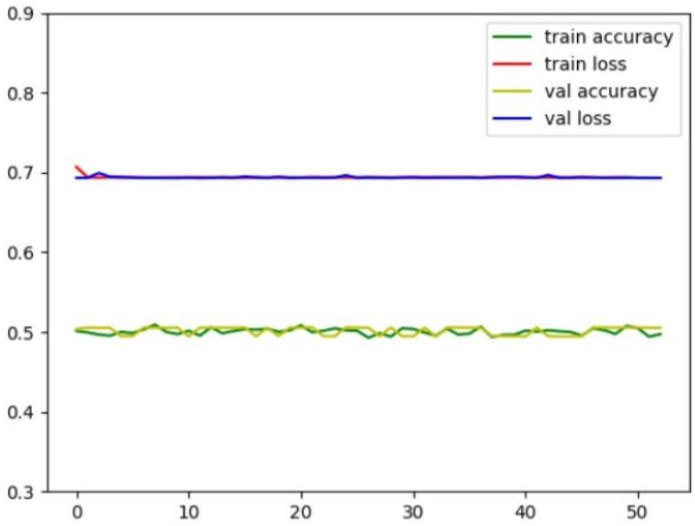


למעשה, כמו בני אדם, וודאי שהמודל יצליח יותר במשימה זו בסרטונים שהם יותר "אינפורמטיביים" ביחס למשימה. למשל – סרטון של אדם המוחא כף, מדבר, או מכה בפטיש, יהיה קל יותר לקבוע האם האודיו והווידאו מסונכרנים. לעומת זאת, בסרטון של מסיבה עם מוזיקת רקע חזקה, יהיה כמעט בלתי אפשרי לקבוע האם ישנו סנכרון או לא.

6.2 דיון וסיכום

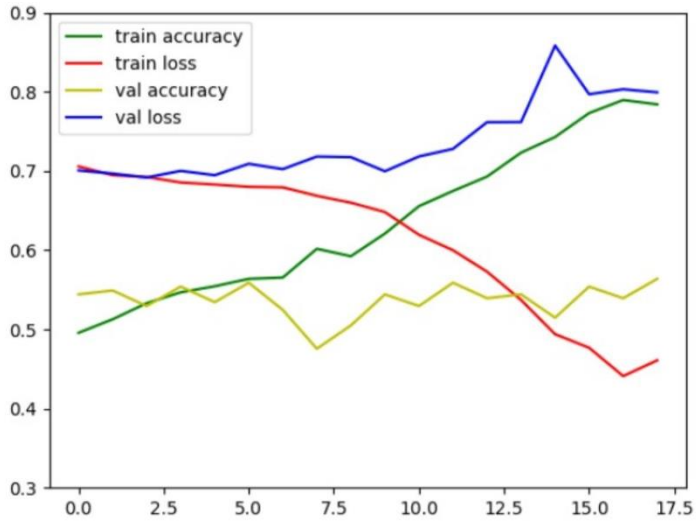
במהלך העבודה על הפרוייקט, עברנו שלבים רבים בעיבוד הדאטה והאימון, אותם נרצה להציג כאן בקצרה.

בתחילת הדרך, המודל שלנו לא הצליח לבצע למידה כלל, והגרפים שקיבלנו נראו כולם כך:



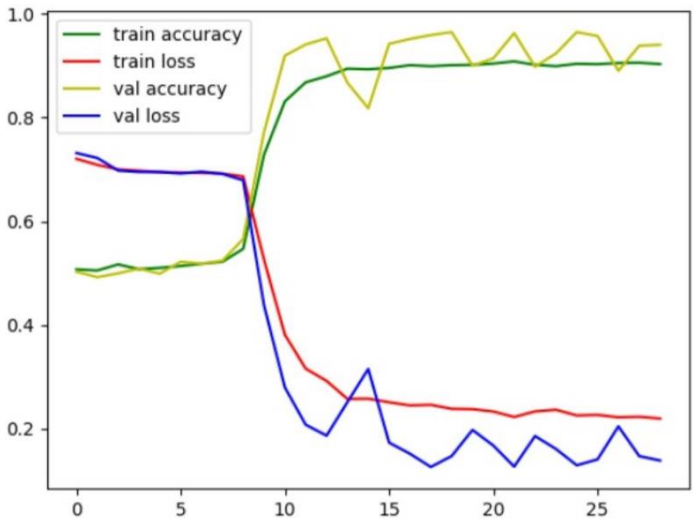
ז"א גם עבור הtrain וגם עבור הוולידציה, ניכר שהמודל לא ביצע שום הנחתה של פונקציית המחיר (loss) ולא הצליח לשפר את רמת הדיוק (accuracy) מעבר לניחוש סתמי בו הוא מצליח ב כ50% מהמקרים.

לאחר הרבה בדיקות, גילינו שהבעיה, ככל הנראה היא שהגדרנו את פרמטר ה weight decay עם ערך גבוה מדי, וכפי שהסברנו בחלק של ההיפר-פרמטרים, הוא לא אפשר למודל לבצע למידה כלל, שכן הוא הגביל מאוד את הגודל שהפרמטרים הנלמדים יכולים לקבל. כאשר סידרנו את הנושא הזה, בהרצה על כמות דאטה קטנה, הצלחנו לקבל גרף למידה המראה overfitting:



כיוון שמדובר על כמות דאטה קטנה, זוהי תוצאה מצוינת המראה שלמודל יש דרך ללמוד, וצריך לבצע אימון על כמות דאטה גדולה יותר ע"מ להגיע לתוצאות טובות גם עבור הוולידציה.

בשלב זה, כאשר עברנו לבצע אימון על כמות דאטה גדולה יותר, קיבלנו תוצאות מפתיעות, אך מפוקפקות:



כאן, גם הוולידציה הגיע לשיעור דיוק לא ייאמן של יותר מ90%. שיעורי הצלחה כאלו נראו לא הגיוניים, במיוחד בהתחשב בעובדה, שבמאמר אותו מימשנו, החוקרים הצליחו להגיע לשיעור דיוק של כ60% בשלב זה. ואכן, במהלך בדיקות של הוויזואליזציה של ה CAM שהתקבלה מתוצאות המודל, ניכר היה שהמודל שם את כל משקל ההחלטה על אות השמע, וליתר דיוק, על 10K הדגימות הראשונות של אות השמע. ז"א שהמודל הצליח למצוא תבנית שנמצאת רק בתחילת אות השמע, לפיה הצליח לקבל החלטה מדויקת להפליא, האם הסרטון מסונכרן או לא. ע"מ להימנע משגיאה זו של המודל, ערכנו אימון נוסף, בו הקלט של המודל זהה, למעט 10K הדגימות הראשונות של אות השמע, אותן איפסנו. בשלב זה קיבלנו תוצאות הגיוניות יותר, שבאו לידי ביטוי בסרטונים עם וויזואליזציה יפה, אפילו עבור אימון על חלק קטן יחסית מהדאטה.

בשלב זה, היינו מוכנים לעבור לאימון המודל הסופי שלנו, על כל הדאטה, אבל דווקא כעת נתקלנו בבעיה טכנית חמורה: מנהלי השרתים של האוניברסיטה שינו את מדיניות הריצה, כך שכל ריצה שלא עומדת בתנאי יעילות מסוימים לאורך זמן – מחוסלת. כיוון שהסתבר לנו שהתוכנה שלנו אינה עומדת בתנאים אלו – נדרשנו לשנות פעמיים את האופן בו אנו מושכים דאטה מהשרתים, ע"מ שפעולה זו לא תיקח זמן רב מדי אשר יגרום לחוסר יעילות בריצה, ולהפסקתה.

כאשר הצלחנו לעמוד בתנאי ההרצה, גילינו שאנו מאמנים מודלים אשר שוב מצליחים ליצור החלטה על סמך אות האודיו בלבד, ולהגיע לתוצאות טובות מבחינת הדיוק.

ע"מ להתגבר על תופעה זו, יצרנו דאטהסט חדש, ובו, במקום ליצור סרטון אחד מכל סרטון מקורי, היכן שאורך הסרטון המקורי היה מספיק – יצרנו שני קליפים קצרים. וכך, בתקווה, מנענו מהמודל למצוא תבניות המסתמכות רק על אות השמע בשלב מסוים בסרטון.

אולם, כאשר המשכנו באימון מודלים על דאטה עם אות שמע שתחילתו מאופסת, גילינו שבשלב מסוים המודל מצליח לקבל החלטה "נכונה" בשיעורי הצלחה גבוהים מאוד, על סמך אות השמע בלבד. לדוגמא נתבונן שוב בגרף הלמידה של המודל הסופי שלנו: בשלב בו מתחילה העלייה בדיוק החיזוי של המודל, זהו השלב בו המודל עובר להסתמך על אות השמע. ראינו זאת באופן וויזואלי כאשר הפקנו את סרטוני ה CAM של המודלים הללו. עבור

מודל שנלקח מאיטרציה 23, ה CAM יצא הגיוני, ז"א שברוב הסרטונים הוא מראה חיווי על איזורים החשודים בלהיות מקורות הקול בסרטונים.

לעומת זאת, עבור מודל שנלקח מאיטרציה 25 וצפונה, ה CAM צובע באופן "אקראי" פריימים שלמים בשלב כלשהו של הקליפ, סימן לכך שהמודל ריכז את תשומת הלב של ההחלטה באות האודיו בלבד.

השאלה למה המודל מתנהג באופן כזה, ואיך הוא מצליח לזהות האם הקליפים מסונכרנים או לא רק לפי אות האודיו, נותרה בגדר תעלומה, ואנו משאירים אותה ב"צריך עיון". זהו המשך מחקר אפשרי לפרויקט זה.

נראה מספר סרטונים לדוגמה שממחישים את ההתנהגות המוזרה:

איטרציה 23	איטרציה 35

בשלב זה הגענו לתוצאות אותן הצגנו בחלק הקודם. כפי שראינו, בעזרת מודל זה ניתן לקבל תוצאות יפות של וויזואליזציה של מקורות קול בסרטונים. כמובן שאת התוצאות הללו ניתן יהיה לשפר בעזרת דאטה גדול, מספר איטרציות גדול יותר fine-tunning, מוצלח יותר להיפר פרמטרים וכן הלאה, אבל בהחלט ניתן לראות את התועלת של שימוש בגישה רב-חושית למשימה זו.

כפי שכתבנו בהתחלה, לפרויקט זה יכולים מספר רב של יישומים בעולם עריכת ווידאו, אבטחה והנגשת מידע.

כמו כן, כיווני המשך לפרויקט זה יכולים להיות מציאת תחומים נוספים הקשורים למשימת של ווידאו-אודיו בהם מודל זה יכול להיות שימושי, כמו הפרדת דוברים בסרטון, הפרדת מקורות קול ע"ג המסך ומחוצה לו, זיהוי משימות וכן הלאה. כמו כן, כיוון מחקר מסקרן אליו שמנו לב במהלך העבודה על הפרויקט, אשר אינו קשור בהכרח ללמידה עמוקה, הינו התבניות שהמודל הצליח לזהות, עבור אות השמע בלבד במיקומים שונים בזמן.

באופן אישי, נרצה לסכם, כי עברנו עם הפרויקט הזה תהליך משמעותי ומלמד. מרמת ידיעה בסיסית מאוד בכל התחום של למידה עמוקה בתחילת השנה, למדנו דרך הרגליים באופן מרחיב ועמוק מאוד, מעשי ותיאורטי כאחד מגוון של שיטות ופרקטיקות בתחום, ועל כך אנו שמחים ומודים.

תם ולא נשלם

7 ביבליוגרפיה

קישור למאמר עליו מבוסס הפרויקט - https://arxiv.org/pdf/1804.03641.pdf

מבוא

CNN - https://arxiv.org/abs/1511.08458

ResNet - https://arxiv.org/abs/1512.03385

Activation functions - https://www.v7labs.com/blog/neural-networks-activation-functions

CAM - https://s3-us-west-2.amazonaws.com/ieeeshutpages/xplore/xplore-ie-notice.html?#footnotes

בסיס נתונים

Kinetics - https://www.deepmind.com/open-source/kinetics

המודל שלנו

Optimizers - https://www.kdnuggets.com/2020/12/optimization-algorithms-neural-networks.html

Learning Rate Schedules - https://towardsdatascience.com/learning-rate-schedules-and-adaptive-learning-rate-methods-for-deep-learning-2c8f433990d1

Weight Decay - https://towardsdatascience.com/this-thing-called-weight-decay-a7cd4bcfccab