



פרויקט גמר

Learning to Separate Object Sounds by
Watching Unlabeled Video

הפרדת שמע ע"י צפייה בסרטונים לא-מתוייגים

מגישי הפרויקט:

בניה לוי - 207139403

מרדכי מוראדי - 208731240

מנחה: יוחאי ימיני

אחראי אקדמי: פרופ' שרון גנות

תוכן עניינים

5	הבעיה שאותה אנו פותרים
5	מבוא ללמידה עמוקה
5	Deep neural Networks – I Artificial Neural Networks
6	:Perceptron
6	תהליך ה forwardpass ל – Perceptron
7	Perceptron ל – Deep Neural Network
7	מה הם hyperparameters?
7	מה הם activation function ומה תפקידם?
8	Sigmoid
8	Softmax
8	ReLU
9	מהם Backpropagation and gradient descent ולמה זה חשוב?
10	אלגוריתמים שונים ל – Gradient descent
10	SGD - stochastic gradient descent
10	SGD with momentum
11	Adam
12	שכבות שונות ב – Deep neural network
12	Fully connected layer
12	Pooling layer
12	Convolutional layer
14	Dropout layer
14	Batch normalization layer
16	אימון הרשת
16	אתחול פרמטרים
16	אתחול ה - hyperparameter
16	Batch size
16	כמות ה - Epochs
17	Learning rate
18	פונקציית שגיאה – Loss
18	חלוקת ה – dataset
18	יחס חלוקת ה – Dataset
20	הבנת הבעיה
20	מבוא
22	השיטה
22	תקציר רצף המערכת

22	הוצאת וקטורי הבסיס
22	התמרת STFT
23	פירוק NMF
24	מערכת למידה לזיהוי אובייקטי-קול – Weakly-Supervised
24	רשת ה- ResNet
25	Deep MIML(Multi-Instance Multi-Label) Network
26	הפרדת וקטורי-בסיס לכל אובייקט
27	הפרדת אובייקט-קול עבור סרטון חדש
28	יישום השיטה
28	Datasets
28	AudioSet-Unlabeled
28	סינון הדאטא
29	Implementation Details
29	הכנת הדאטא
30	בניית הרשת
31	בניית Custom Data Generator
31	תוצאות
31	Train and Val Loss
32	מפות יחס בסיס – אובייקט
32	כלב וכינור
33	כינור וחליל
34	גיטרה אקוסטית ו- Banjo
35	הצעות לשיפור
36	מסקנות
37	Bibliography

רשימת איורים

5	איור 1: רשת נירונים "רדודה"
5	איור 2: רשת נירונים עמוקה
6	איור 3: Perceptron
6	איור 4: MLP
7	איור 5
8	איור 6: common activation functions
8	איור 7: sigmoid function
9	איור 8: ReLu function
10	איור 9: backpropogation algorithm
9	איור 10: gradient descent ברשת נירונים
10	איור 11: SGD with and without momentum
10	איור 12: תיאור מתמטי ל- SGD with momentum
11	איור 13: תיאור מתמטי ל- Adam

12	fully connected layer:	איור 14
12	example of pooling layer	איור 15
13	החלקת המסנן והכפלה איבר איבר	איור 16
13	convolutional layer –	איור 17
14	Dropout	איור 18
15	חיסור הממוצע מהקלט לרשת	איור 19
15	נרמול הקלט לרשת	איור 20
15	Batch normalization layer –	איור 21
16	שגיאה כתלות במספר ה – epoch-ים שהרשת אומנה	איור 22
17	ירידת ה Loss בשלושה קצבי למידה שונים	איור 23
17	השפעת קצב הלמידה על קצב ירידת ה Loss	איור 24
18	Classification and Regression Loss	איור 25
19	DataSet – פיצול ה	איור 26
20	הפרדת השמע לצלילים המרכיבים אותו	איור 27
22	רצף מערכת הלמידה	איור 28
23	תיאור מתמטי להתמרת STFT ו ISTFT	איור 29
23	NMF שבוצע על הספקטרום של רצף קצר של פסנתר שמורכב מחמישה תווים	איור 30
24	בלוק בודד ברשת ResNet	איור 31
24	תיאור מתמטי להוספת הגראדינט ברשת ה – ResNet	איור 32
25	מבנה רשת ה – MIML	איור 33
26	Hinge Loss כאשר x הוא חיזוי הרשת ו y וקטור מאפס עד גודלו של x	איור 34
26	דוגמאות של basis – object relation map	איור 35
28	תרשים זרימה לשלב ה - test	איור 36
29	סינון מערך הנתונים	איור 37
30	מבנה רשת ה – MIML	איור 38
31	Loss של האימון והולידציה	איור 39
32	מפה מאיפוק 20. משמאל – מפה מאיפוק 140	איור 40
32	מפת יחס בסיס – אובייקט, שמאל – פריים מהסרטון	איור 41
33	מפת יחס בסיס – אובייקט, שמאל – פריים מהסרטון	איור 42
34	מפת יחס בסיס – אובייקט, שמאל – פריים מהסרטון	איור 43

הבעיה שאותה אנו פותרים

הבעיה שבפריקט זה באנו לפתור וליישם היא הפרדת דיבור באמצעות מידע חזותי וערוץ מיקרופון בודד.

כידוע התבוננות בסצנה בצורה מלאה דורשת את כל החושים. כלומר מידול של מראה וצליל של אובייקטים, מאתגרת: רוב הסצנות והאירועים הטבעיים מכילים מספר עצמים, ורצועת האודיו מערבבת את כל מקורות הסאונד יחד.

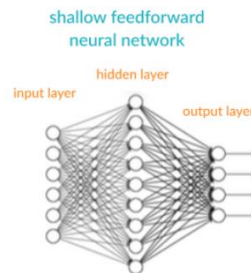
אנו מציעים ללמוד מודלים של אובייקטי audio-visual (משלבים ראייה ושמיעה) מווידיאו לא מתויג, ואז להשתמש בהקשר החזותי כדי לבצע הפרדת מקור שמע בסרטוני וידאו חדשים. גישתנו מסתמכת על רשת נירונים MIML כדי להפריד את בסיסי תדרי השמע הממפים לאובייקטים חזותיים בודדים, אפילו מבלי להתבונן / לשמוע אותם עצמים ביחידות. אנו מראים כיצד ניתן להשתמש בבסיסים המופרדים כדי להנחות את ההפרדה במקור השמע וכך להשיג צלילים המופרדים טוב יותר ברמת האובייקט.

מבוא ללמידה עמוקה

בחלק זה נסביר מושגים בסיסיים שנעשה בהם שימוש לאורך הפריקט:

Deep neural Networks – I Artificial Neural Networks

רשתות נירונים (ANN) היא מערכת למידה מפוקחת הבנויה ממספר גדול של אלמנטים פשוטים, המכונים נירונים. כל נירון יכול לקבל החלטות פשוטות, ומזין את ההחלטות הללו לנירונים אחרים, המאורגנים בשכבות מחוברות זה לזה. יחד, הרשת העצבית יכולה להביע כמעט כל פונקציה, ולענות כמעט על כל שאלה (למשל סיווג), בהינתן מספיק דגימות אימון וכוח מחשוב. ברשת נירונים "רדודה" יש רק שלוש שכבות של נירונים:



איור 1: רשת נירונים "רדודה"

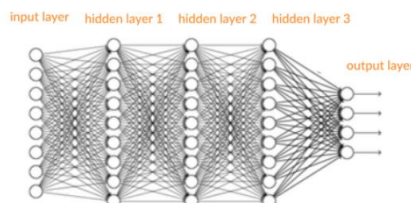
Input layer - מקבלת את ה - training data של המודל שאותו נרצה לאמן.

Hidden layer - שכבה אחת נסתרת

Output layer - שכבת פלט שמייצרת את החיזויים.

לרשת נירונים עמוקה (DNN) יש מבנה דומה, אך יש לה שתי "שכבות נסתרות" או יותר של נירונים. הראו כי בעוד שרשתות נירונים רדודות מסוגלות להתמודד עם בעיות מורכבות, רשתות למידה עמוקה מדויקות יותר ומשתפרות ברמת הדיוק ככל שמתווספות עוד שכבות נירונים. כיום רוב מודלי רשת הנירונים והיישומים בהם משתמשים ברשת עמוקה של בין 3-10 שכבות נירונים.

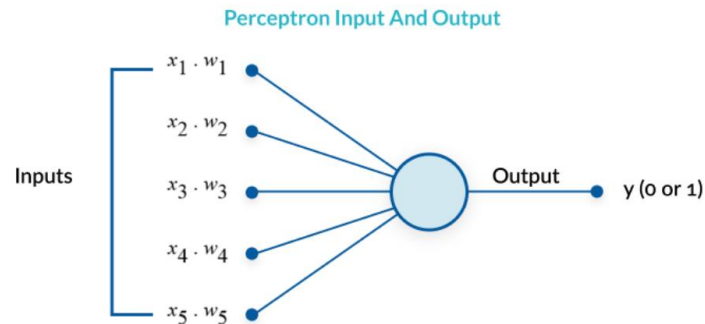
Deep neural network



איור 2: רשת נירונים עמוקה

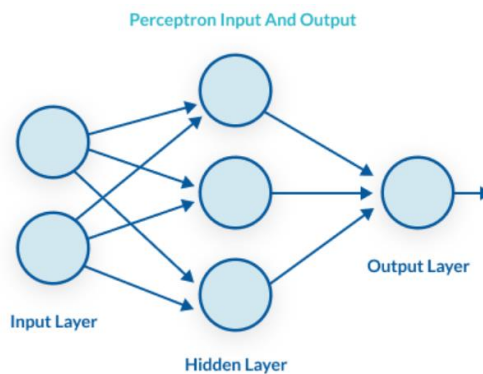
Perceptron:

Perceptron הוא יחידת סיווג בינארי שעוצבה על פי תפקוד המוח האנושי - היא נועדה לחקות את הנירון. למרות שיש ל - perceptron מבנה פשוט, הוא מרכיב ייסודי למערכות שיכולות ללמוד ולפתור בעיות מורכבות מאוד.



איור 3: Perceptron

כך נוכל לשלב קבוצה של perceptrons בכל שכבה כפי שמוצג באיור 4



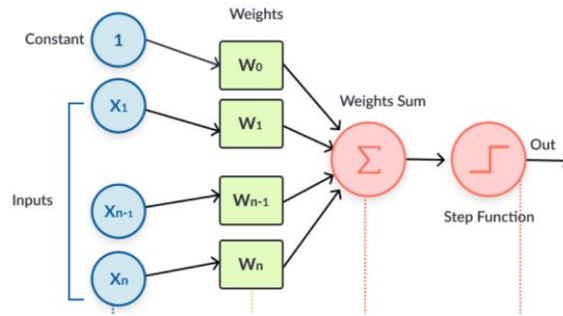
איור 4: MLP

תהליך ה - forwardpass ל - Perceptron:

תהליך ה - forwardpass של ה - perceptron מתבצע באופן הבא:

1. הכניסות שמוזנות אל תוך הרשת מוכפלות במשקולות המתאימות ומתבצעת סכימה.
2. מוסיפים לסכום קבוע שנקרא bias ("משקל הטיה") שמטרתו להזיז את פונקציית הפלט של כל perceptron לכיוון מסוים בערכו.
3. הסכום מוזן דרך activation function.
4. פלט ה - activation function מהווה את המוצא של ה - perceptron.

תהליך למידה זה ניתן לראות באיור 5 (כאשר פונק' המדרגה מהווה את ה - activation function)



איור 5

מ – Perceptron ל – Deep Neural Network:

לאחר שהבנו מהם perceptrons נסביר עוד כמה דברים שמרכיבים אותה, בשביל להבין את מבנה הרשת ה-DNN כולה ופעולותיה: hyperparameters ועוד סוגי activation functions.

מה הם hyperparameters?

hyperparameters –רשת הניורונים גמישה יותר מבחינת תהליך הלמידה בעקבות ה - hyperparameters שהם פרמטרים חיצוניים שנקבעים ע"י מפעיל הרשת כמו מספר איטרציות אימונים (iteration and epoch), אתחול המשקולות, רגולריזציה ועוד, כפי שניתן לראות בטבלה הבאה:

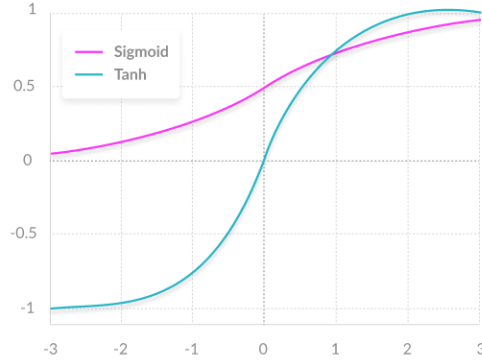
Hyperparameters related to the training algorithm	Hyperparameters related to neural network structure
<ul style="list-style-type: none"> • Learning rate • Epoch, iterations and batch size • Optimizer algorithm • Momentum 	<ul style="list-style-type: none"> • Number of hidden layers • Dropout • Activation function • Weights initialization

Advanced architectures - רשתות ניורונים יכולות להכיל מגוון ארכיטקטורות שיכולות לעזור בפתרון בעיות ספציפיות. כמו למשל RNN, CNN, GAN כאשר אנו נעשה שימוש בשתיים הראשונות.

Activation function –רשת ניורונים משתמשת במגוון activation functions המפיקות ערכים אמיתיים, לא ערכים בוליאניים כמו ב - perceptron הקלאסי המשתמש בפונקציית מדרגה

מה הם activation function ומה תפקידם?

activation function היא פונקציה שקובעת את המוצא של כל אלמנט (perceptron or neuron) ברשת ניורונים. רשת ניורונים משתמשת במגוון activation functions המפיקות ערכים רציפים או ערכים בוליאניים. הפונקציה לוקחת את הקלט מכל ניורון והופך אותו לפלט, בדרך כלל בין אחד לאפס או בין 0 ל-1. ל – 1 כפי שניתן לראות באיור 6. Classic activation functions המשמשות ברשתות ניורונים הם פונקציית מדרגה (שיש לה מוצא בינארי), sigmoid ו- tanh. activation functions חדשות, שנועדו לשפר את היעילות החישובית, הם ReLU.



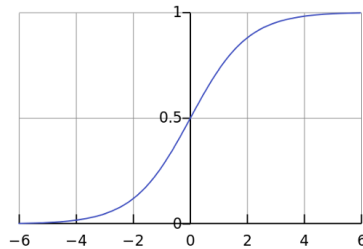
איור 6: common activation functions

פרט את ה- activation functions שלכל אחד תפקיד משלו:

Sigmoid

כפי שניתן לראות באיור 7: sigmoid function לפונקציה זו המוצא יהיה בין אפס לאחד, יתרונותיה של פונקציה זו הם חוסר הליניאריות ובנוסף אם נשתמש בפונקציה זו ברצף לאורך השכבות ברשת שלנו נקבל תמיד ערך בין אפס לאחד בשונה מפונקציה ליניארית שבה יש שני בעיות, אחת שנוכל לקבל ערכים בטווח $(-\infty, \infty)$ וזה גורם לשונות בין משקלים גבוהה מאוד ולכן הלמידה תהיה איטית יותר. והשנייה, שהיא תיתן רשת ליניארית ולכן פחות expressiveness. אולם קיימת בעיה בפונקציה זו משום שעבור ערכים גבוהים או קטנים מאוד של פרמטרי הקלט לפונקציה, הנגזרת קרובה מאוד ל-0 אז רשת שמבוססת על למידה על פי הנגזרות תלמד בצורה מאוד איטית, בעיה זו נקראת vanishing gradient.

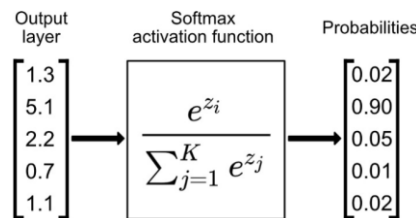
$$A = \frac{1}{1+e^{-x}}$$



איור 7: sigmoid function

Softmax

פונקציה זו מנרמלת את המוצאים עבור כל מחלקה (class) בין אפס לאחד וכך מחזירה את ההסתברות שהקלט הנכנס לרשת שייך למחלקה מסוימת.



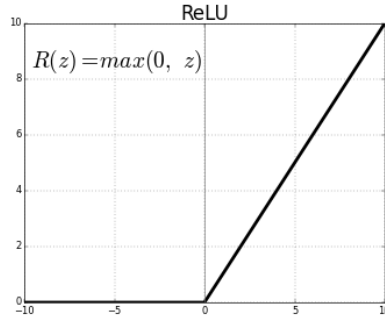
ReLu

כפי שניתן לראות באיור 8, במבט ראשון על פונקציה זו, נראה כאילו יש את הבעיה של התפקוד הליניארי שכן פונקציה זו ליניארית בציר החיובי, אולם הפונקציה כולה אינה ליניארית ולכן נוכל להשתמש בה ברצף לאורך השכבות ברשת שלנו. יתרון נוסף של פונקציה זו היא הדלילות שהיא מכניסה לרשת, וזאת משום

שבציר השלילי של הפונקציה יש ערך אפס וכך אנו גורמים לחלק מהנירונים להיות "מושתקים", not activate, ובכך נגרום ליעילות ברשת.

אולם משום שפונקציה זו מקבלת ערך אפס בציר השלילי אנו יכולים לקבל שבנירונים מסוימים לא תתרחש עדכון משום שה – gradient במקומות אלו יהיה אפס ובכך ניירונים שעברו למצב זה (נכנסו לאזור הציר השלילי של פונקציית ReLU) יפסיקו להגיב לשינויים בשגיאה, כלומר לא יתרמו להקטנת פונקציית השגיאה, בעיה זו נקראת dying ReLU.

מבחינה חישובית, פונקציה זו פחות יקרה מ – sigmoid מכיוון שהיא כוללת פעולות מתמטיות פשוטות יותר, זו נקודה חשובה שיש לשים לב אליה כאשר בונים רשתות ניירונים עמוקות.

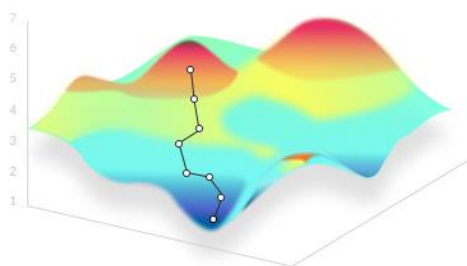


איור 8: ReLU function

מהם Backpropagation and gradient descent ולמה זה חשוב?

לאחר אתחול רשת הנירונים עם משקולות ראשוניות, וביצוע forward pass בכדי לייצר את החיזוי הראשוני, קיימת פונקציית שגיאה המגדירה כמה רחוק המודל מהערך אותו רצינו שהרשת תחזה. ישנם אלגוריתמים רבים שיכולים למזער את פונקציית השגיאה - לדוגמה, ניתן לבצע מינימיזציה וגזירה על פונקציית השגיאה כדי למצוא את המשקולות שיוצרים את השגיאה הקטנה ביותר. עם זאת, עבור רשתות ניירונים גדולות יש צורך באלגוריתם שיהיה יעיל יותר חישובית. לצורך כך קיים אלגוריתם ה – Backpropagation, הוא יכול לגלות את המשקולות האופטימליות יחסית מהר אפילו עבור רשת עם מיליוני משקולות.

זה gradient descent אלגוריתם אופטימיזציה שנועד לצמצם פונקציה מסוימת ע"י תזוזות איטרטיביות בכיוון הירידה התלולה ביותר שהיא בכיוון השלילי של הגרדיאנט.



איור 9: gradient descent ברשת ניירונים

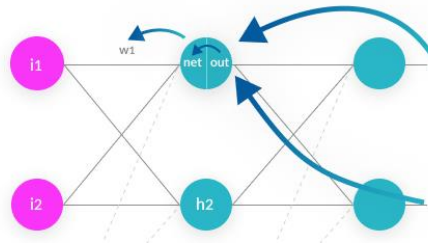
תהליך ה gradient descent מתבצע באופן הבא:

1. Forward pass – אתחול המשקולות והזנת ה – training data אל תוך הרשת, יצירת חיזוי.
2. Backpropagation with gradient descent – אלגוריתם זה מחשב עד כמה מושפעים ערכי החיזוי מכל אחד מהמשקלים במודל. לשם כך אלגוריתם זה מחשב נגזרות חלקיות, החל מפונקציית

השגיאה עד לנוירון ספציפי ומשקלו באמצעות כלל השרשרת: $\frac{\partial z}{\partial x}|_x = \frac{\partial z}{\partial y}|_{y(x)} \cdot \frac{\partial y}{\partial x}|_x$ בדרך זו נוכל

לעקוב ולראות כיצד כל נוירון ומשקלו משפיע על פונקציית השגיאה הכללית. כך נקבל מערכת משקולות שמצמצמות את פונקציית השגיאה.

3. Weight update – עדכון המשקולות. אפשר לבצע את עדכון המשקולות לאחר כל הדוגמאות שבדטא אך בד"כ זה לא מעשי. שיטה אחרת זה לעדכן לאחר כל דוגמא מה – training data, אך בדרך כלל זאת לא שיטה טובה לאימון הרשת, כי כך הלמידה תתמקד בשיפור הדוגמה הספציפית בכל פעם ודוגמה אחת, בד"כ פחות מייצגת את כלל הדטא שקיים. ולכן בדרך כלל, מה שעושים זה מגרילים בכל איטרציה קבוצה של דוגמאות (batch) מסט האימון, ועליה מפעילים gradient descent באמצעות ה – backpropogation כאשר גודל - batch, מספר האיטרציות וה – epoch נקבע כ – hyperparameters כפי שציינו.



איור 10: backpropogation algorithm

אלגוריתמים שונים ל – Gradient descent

נפרט יותר על שלב ה – gradient descent - לאחר שחישבנו את הנגזרות החלקיות, נרצה לעדכן את המשקולות כך שנגיע לנקודת מינימום בפונקציית השגיאה.

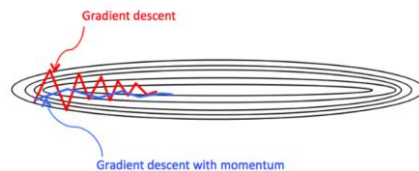
אפשר לעשות זאת בכמה שיטות שונות:

SGD - stochastic gradient descent

תהליך עדכון המשקולות מתבצע בצורה פשוטה על סמך ה – batch הנוכחי שנכנס לרשת בלי התחשבות ב – batch-ים קודמים

SGD with momentum

לעדכון המשקולות המתואר בסעיף 1 יש חסרון כאשר אנו נמצאים באזור שבו בציר אחד השינוי הוא גדול ובציר השני השינוי קטן, כתוצאה מכך נקבל את תופעה המתוארת באיור 11 באדום. כדי להתגבר על בעיה זו נרצה להשתמש במומנטום שיש לנו בתהליך הלמידה וכך נקבל את המתואר באיור 11 בכחול.



איור 11: SGD with and without momentum

המומנטום מאיץ את ה – SGD בכיוון הרלוונטי ובכך מונע את האוסילציות הללו. התיאור המתמטי לשיטה זו נתון באיור 12.

$$V_t = \beta V_{t-1} + (1 - \beta) \nabla_w L(W, X, y)$$

$$W = W - \alpha V_t$$

איור 12: תיאור מתמטי ל – SGD with momentum

זוהי שיטה המשלבת בתוכה את שני השיטות:

AdaGrad המתאים קצב למידה לכל פרמטר ברשת הניורונים ומשפר את הביצועים בבעיות שהגראדינט "חלש" (sparse gradient), RMSProp המתאים גם קצב למידה לכל פרמטר ברשת הניורונים וזאת ע"פ ממוצע אמפליטודות הגראדינט האחרונות עבור כל פרמטר. בשיטה זו נצליח בבעיות שהן מקוונות (למשל רועשות). Adam משלב את יתרונות שני השיטות לעיל. Adam מתחשב בממוצע המומנט השני (השונות) בשונה מ-RMSProp המתאים את קצב למידת הפרמטרים על סמך ממוצע אמפליטודות הגראדינט שזהו המומנט הראשון (הממוצע). את התיאור המתמטי לשיטה זו ניתן לראות באיור 13.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

א - moving average על הגראדינט והגראדיאנט בריבוע

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

ב - משערכי מהמומנט הראשון והשני לאחר נרמול

$$w_t = w_{t-1} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}$$

ג - עדכון המשקולת

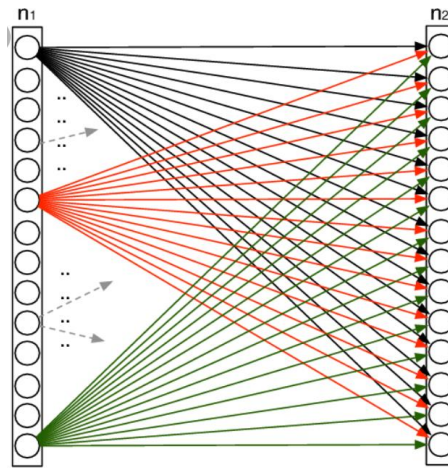
איור 13: תיאור מתמטי ל-Adam

בנוסף ליתרונות שהוצגו, ל-Adam יתרון נוסף, גודל צעד העדכון (eta) בלתי משתנה כתלות במגניטודה של הגראדינט, דבר אשר יכול לעזור לנו במקרים של saddle point and ravines (ל-SGD יהיו פיתולים כפי שמתואר באיור 11).

חשוב לזכור שלפעמים שיטת עדכון זו, אינה הכי אידיאלית ולא מתכנסת לפתרון האופטימלי כך למשל עבור סיווג תמונה כאשר CIFAR מהווה את מאגר המידע, הפתרון האופטימלי עבור בעיה זו היא עם שיטת ה-SGD+momentum.

Fully connected layer

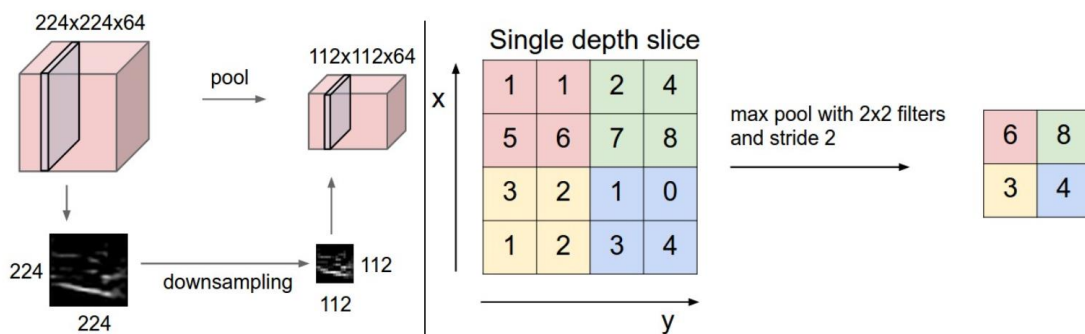
שכבה בה כל הכניסות אליה מחוברות לכל activation unit של השכבה הקודמת, כפי שמתואר באיור 14. בהרבה מודלים, השכבות האחרונות הן שכבות fully connected, המשלבות את הנתונים שחולצו על ידי השכבות הקודמות כדי ליצור את הפלט הסופי (למשל סיווג המידע ב – class-ים שונים). החיסרון בוא שהוא בזבזני ולעיתים לא כל ניורון במוצא דורש חיבור לכל ניורון שבכניסה.



איור 13: fully connected layer

Pooling layer

שכבה שנועדה לצמצם בהדרגה את הגודל המרחבי של הייצוג ברשת (העומק נשאר זהה) וזאת כדי להפחית את כמות הפרמטרים והחישוב ברשת, בנוסף לכך, שכבה זו גורמת לרשת להיות יותר גנרית משום שבשכבה זו מתבצע איחוד של מספר ערכים אל ערך אחד ובכך משהו ייחודי עבור מידע מסוים יעלם בשכבה זו ומכאן נפתור את בעיית ה – overfitting. שכבת זו פועלת באופן בלתי תלוי על כל פרוסת עומק של הקלט ומשנה את גודלו המרחבי תוך שימוש בפעולת מקסימום או ממוצע כפי המתואר באיור 15.



איור 14: example of pooling layer

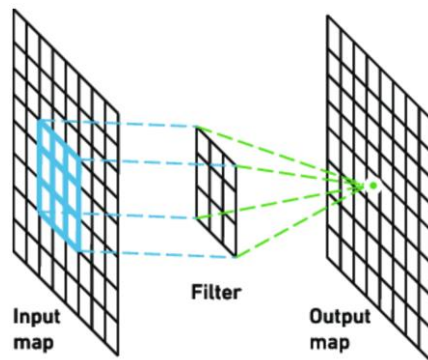
Convolutional layer

הפרמטרים של שכבת ה- convolution מורכבים מסט פילטרים הניתן ללמידה. כל פילטר הוא קטן יחסית באופן מרחבי **המשתרע לאורך העומק** של הקלט. כך לדוגמה, מסנן טיפוסי בשכבה ראשונה של ה – ConvNet בעל ממדים 5x5x3, כלומר רוחב וגובה של 5 פיקסלים, וה- 3 מכיוון שלתמונות עומק 3, ערוצי הצבע. במהלך ה – forward-pass, אנו מחליקים כל פילטר לרוחב ולגובה של נפח הקלט, מבצעים מכפלת

איבר-איבר של כל המסנן בכל מיקום בקלט כמתואר באיור 16. כאשר מחליקים את המסנן לרוחב ולגובה של נפח הקלט, נוצר activation map המתאר את התגובות של אותו פילטר בכל מיקום מרחבי. הרשת לומדת את הפרמטרים של הפילטרים המופעלים כאשר הם רואים סוג כלשהו של תכונה חזותית כמו קצוות, כתם של צבע כלשהו בשכבה הראשונה, או תכונות שמתאימות לשכבות גבוהות יותר של הרשת. כך נקבל סט שלם של פילטרים בכל שכבת convolution, שכל אחד מהם יפיק activation map נפרדת. לאחר מכן נחבר לאורך ממד העומק את ה- activation maps הללו ונקבל את המוצא המתאים.

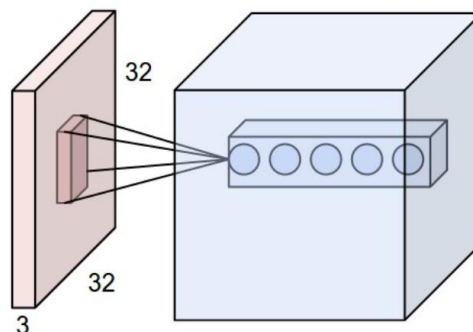
נקודות נוספות אודות המבנה המרחבי של הפלט:

- מספר ערוצי המוצא הוא hyper-parameter: מספר המסננים שנרצה להשתמש בהם מהווים את מספר ערוצי המוצא, כאשר כל אחד מהמסננים הללו מנסה לחפש משהו שונה בקלט, כמו כן גם גודל הפילטר הוא hyper-parameter.
- עלינו לציין את הצעד בו אנו מחליקים את המסנן. כאשר הצעד הוא 1 אז אנו מזיזים את המסננים פיקסל אחד בכל פעם. כאשר הצעד הוא 2 אז המסננים קופצים 2 פיקסלים בכל פעם שאנחנו מחליקים אותם, צעד אשר יגרום להקטנת נפח המוצא במרחב.
- לפעמים נוח לרפד את נפח הקלט באפסים סביב הגבול. הגודל של הריפוד באפסים הוא hyperparameter. הריפוד באפסים מאפשר לנו לשלוט על הגודל המרחבי של נפחי הפלט. לעיתים קרובות נשתמש בריפוד זה כדי לשמר במדויק את הגודל המרחבי של נפח הקלט כך שרוחב וגובה הקלט יהיו זהים לרוחב וגובה הפלט בהתאמה.



איור 15: החלקת המסנן והכפלה איבר איבר

דוגמה לשכבת ה- convolution ניתן לראות באיור 17: כל נירון בשכבת ה- convolutional קשור לאזור מקומי בנפח הקלט במרחב לכל העומק (3 ערוצי צבע בתמונה צבעונית), ניתן לראות שקיימים 5 נירונים המתאימים לאותו אזור, זהו מספר המסננים שאנו מפעילים על אזור זה אשר בודק תכונות נוספות ואחרות בתמונה כפי שציינו לעיל.



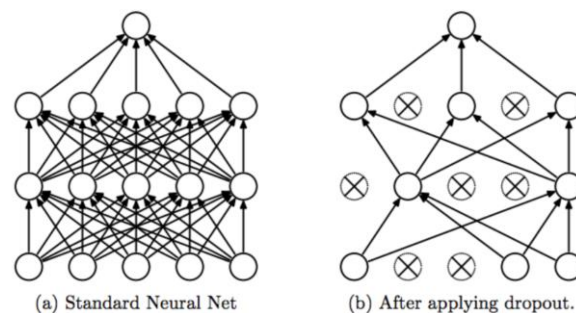
איור 16: דוגמה ל- convolutional layer

Dropout layer

שכבה המאפסת נירונים באופן באקראי בשלב האימון וכך מונעת מהם לעדכן את המשקולות שלהם כפי שמתואר באיור 18. מבחינה טכנית, בכל שלב באימון, נושרים (drop) צמתים מסוימים מהרשת עם הסתברות של $1-p$ או שהם נשארים עם ההסתברות p , כך שהרשת המתקבלת קטנה יותר. בשלב ה- test כל הנירונים קיימים ופועלים, אולם, מנרמלים אותם בפקטור p כדי לפצות על כך שבשלב האימון חלקם לא פעלו.

השימוש בשכבה זו הוא בדרך"כ על שכבת ה- fully connected, אולם, אפשרי להשתמש בשכבה זו גם אחרי שכבת ה- pooling.

מטרת שכבה זו היא למנוע את בעיית ה- overfitting: איפוס נירונים באופן אקראי בשלב האימון מצריך שניירונים אחרים יצטרכו להתמודד עם הייצוג הדרוש כדי ליצור את החיזוי של הנירונים החסרים, בכך, הרשת נהיית פחות רגישה למשקולות ספציפיים של נירונים וכתוצאה מכך, גדלה היכולת של הרשת להכללה של מידע נוסף מעבר ל- dataset של האימון וקטנה הסבירות ל- overfit של ה- training data.

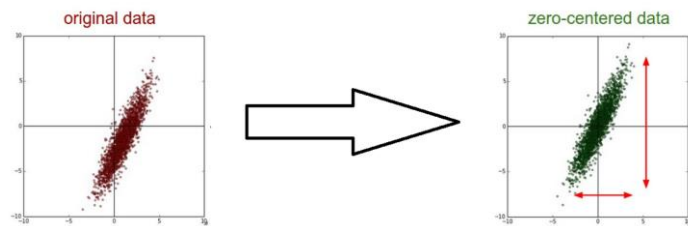


איור 18: Dropout

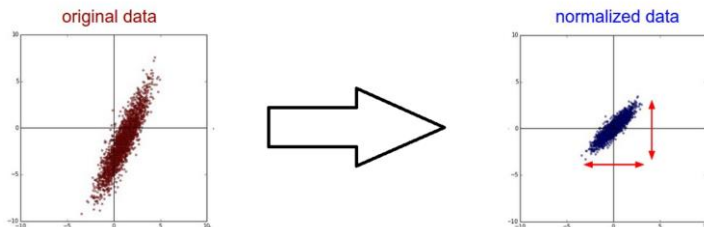
Batch normalization layer

שכבה זו אנו מנרמלים את הקלט לשכבה וזאת ע"י התאמת הסטטיסטיקה של האקטיבציות (מוצא השכבה שלפני). לדוגמה, כשיש לנו אקטיבציות שערכם מ-0 ל-1 וחלקם מ-1 עד 1000, עלינו לנרמל אותם כדי שהרשת תוכל ללמוד באופן מהיר יותר. מסיבה זו מבצעים נרמול לא רק על הקלט שנכנס לרשת אלא גם על הערכים של ה- hidden layers שהם משתנים כל הזמן ובכך נקבל שיפור פי 10 או יותר במהירות אימון הרשת.

בנוסף, שכבה זו מקטינה את בעיית ה- covariance shift. בעיה זו מתייחסת לשינוי בהתפלגות בערכי הקלט לרשת הנירונים. זו בעיה שאינה ייחודית דווקא ל- deep learning. לדוגמה, אם ה- dataset של ה- train and test מגיעים ממקורות שונים לחלוטין (למשל תמונות ה- train מגיעות מהאינטרנט בזמן שתמונות ה- test הן תמונות שצולמו מהפלאפון), ההתפלגות של שני ה- dataset הללו יהיו שונות. בהקשר של למידה עמוקה, אנו חוששים מהשינוי בהתפלגות של הקלט לרשת משום שברשת נירונים המשקולות בכל שכבה משתנים לאורך שלב האימון, ומכך גם האקטיבציות. מכיוון שהאקטיבציה של שכבה קודמת היא כניסת השכבה הבאה, כל שכבה ברשת הנירונים מתמודדת עם מצב בו התפלגות הקלט משתנה בכל שלב. זהו מצב בעייתי מכיוון שהוא מאלץ כל hidden layers להסתגל ברציפות לשינוי הכניסות. כפתרון לבעיה זו קיים שכבת ה- batch normalization, הרעיון הבסיסי של שכבה זו הוא להגביל את ה- covariance shift על ידי נרמול האקטיבציות של כל שכבה (ממוצע אפס ושונות אחד, כפי שניתן לראות באיור 19-20 עבור הקלט לרשת). דבר זה מאפשר לכל שכבה ללמוד על התפלגות יציבה יותר של הקלט, וכך להאיץ את אימון הרשת.



איור 19: חיסור הממוצע מהקלט לרשת



איור 20: נרמול הקלט לרשת

אולם, הגבלת התפלגות האקטיבציה של כל שכבה להיות ממוצע אפס ושונות 1 יכולה להגביל את הפוטנציאל של הרשת. לכן בפועל, batch normalization מאפשרת לרשת ללמוד פרמטרי γ and β שיכולים להמיר את הממוצע והשונות לכל ערך שהרשת רואה לנכון. ניתן לראות את התיאור המתמטי באיור 21.

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;

Parameters to be learned: γ, β

Output: $\{y_i = \text{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^m x_i \quad // \text{ mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{ mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{ normalize}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma,\beta}(x_i) \quad // \text{ scale and shift}$$

איור 17: תיאור מתמטי ל – Batch normalization layer

לכן היתרונות לשימוש בשכבה זו הם:

- אפשרי להשתמש בקצב למידה גבוה מכיוון שהנורמליזציה גורמת לכך שלא יהיו אקטיבציות שיהיו גבוהות או נמוכות מידי. לכן, רשת שלא יכולה לאמן לפני, יתאפשר כעת.
- שכבה זו מפחיתה את בעיית ה – overfitting מכיוון שיש לה השפעות רגולריזציה. בדומה ל – dropout, הנורמליזציה מוסיפה רעש לאקטיבציות של ה – hidden layers. לכן, אם אנו משתמשים ב – batch normalization, אזי פחות נשתמש ב – dropout, ובכך לא נאבד מידע רב. אולם, לצורך הרגולריזציה מומלץ להשתמש ב – batch normalization ביחד עם dropout ולא להיות תלוי רק בנורמליזציה.

אימון הרשת

לאחר שהסברנו את בניית הרשת באמצעות הרכיבים שנאמרו לעיל, נוכל לדבר על האימון של הרשת. לצורך כך נצטרך להסביר עוד שני דברים: אתחול הפרמטרים, אתחול hyperparameter, הגדרת פונקציית השגיאה (Loss).

אתחול פרמטרים

רשתות נוירונים עמוקות מכילות מיליונים או מיליארדי פרמטרים. דרך האתחול של פרמטרים אלו יכולה לקבוע כמה מהר הלמידה שלנו תתכנס וכמה מדויקת היא תהיה. הדרך הישירה היא לאתחל את הפרמטרים לאפס. עם זאת, אם אנו מאתחלים את המשקולות של כל שכבה לאפס, הגראדינט יהיה זהה לכל משקולות בשכבה ולכן העדכון למשקולות יהיה זהה לכל הפרמטרים באותה שכבה. לכן אין ספק שאנחנו יכולים לקבל תוצאה טובה יותר באתחול המשקולות עם מספרים אקראיים קטנים.

אתחול ה - hyperparameter

בנוסף לפרמטרים לעיל, נרצה לאתחל באופן מושכל את ערכי ה - hyperparameters.

Batch size

בלמידת מכונה בכלל ובלמידה עמוקה בפרט ישנו trade-off בין הדיוק למהירות.

כאשר נבחר את הגודל ה - batch להיות 1 או מספר קטן מאוד אחר, אזי יהיה אלמנט משמעותי של רעש בתהליך העדכון שמתבצע ע"פ הגראדינט משום שזה מתבסס על מידע קטן מאוד. אולם מכיוון שמשמשים במידע קטן, אז הרבה יותר מהיר לחשב את תהליך העדכון, forward and backward pass ומכאן להשיג את השיפוע. לכן גודל batch של אחד הוא מהיר אך לא מדויק.

מצד שני, כאשר נבחר גודל batch גדול מאוד אזי התהליך ייקח יותר זמן, אך יהיה מדויק יותר מכיוון שהשונות יורדת כאשר נמצע את דגימות.

לכן בתחילת תהליך האימון של הרשת אנו ננסה כמה ערכים מושכלים ונראה מה מניב את התוצאה הטובה ביותר.

ברוב המקרים נבחר את גודל ה - batch להיות חזקה של 2, זה נובע מאופן קידוד ה-gpu, כאשר הגודל הוא חזקה של 2 ה-gpu מסוגל לחלק את עומס העבודה בצורה טובה יותר (threads).

כאשר הרשת עוברת על כל ה dataset (על כל ה batches) זה נקרא שהיא סיימה epoch.

כמות ה - Epochs

ככל שמספר ה-epoch גדל, מספר הפעמים שנבצע עדכון למשקלים ברשת הנוירונים גדלה גם היא. בכך התנהגות הרשת תהיה: underfitting → optimal → overfitting כפי שניתן לראות באיור 22.



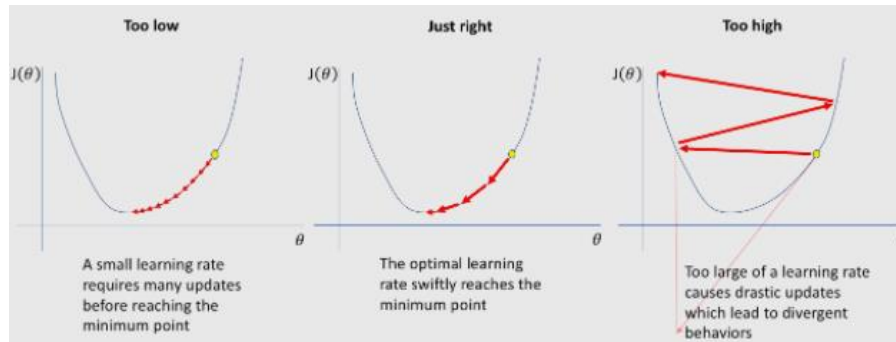
איור 18: שגיאה כתלות במספר ה - epochים שהרשת אומנה

ובכן, גודל מס ה- epoch אינו משמעותי כל כך. חשוב יותר לבדוק את שגיאת ה- training and validation data. כל עוד שתי השגיאות הללו ממשיכות לרדת, נמשיך לבחון epoch גדול יותר עד שנגיע לנקודת האופטימלית.

Learning rate

אחד מה- hyperparameter המרכזיים שיש להגדיר בשביל לאמן את הרשת הוא ה- learning rate. פרמטר זה מגדיל את גודל עדכוני המשקל על מנת למזער את פונקציית השגיאה (Loss) של הרשת.

קצב למידה נמוך מדי, האימונים יתבצעו לאט מאוד משום שהעדכונים למשקלים יהיו זעירים מאוד. על אותו אופן אם קצב הלמידה גבוה מדי, זה עלול לגרום להתנהגות בלתי רצויה של פונקציית השגיאה (Loss) כפי שניתן לראות באיור 23.

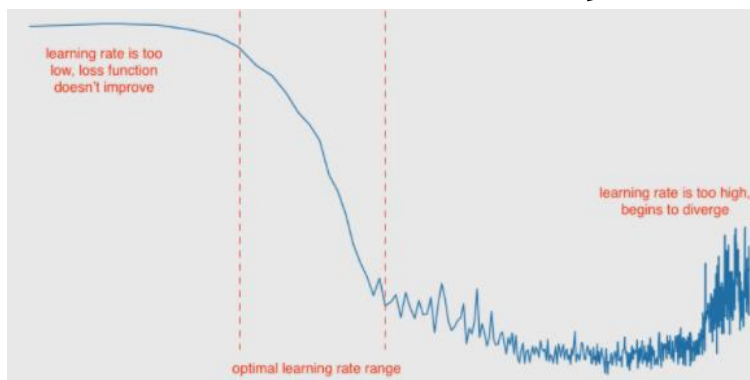


איור 23: ירידת ה Loss בשלושה קצבי למידה שונים

קצב הלמידה האופטימלי תלוי בטופולוגיה של ה Loss, כאשר הטופולוגיה תלויה בארכיטקטורת המודל וב dataset שנשתמש בו לצורך האימון. שימוש בקצב למידה על פי ברירת המחדל (כלומר ברירת המחדל שנקבעה על ידי הספריות שמשמשות) עשויה להניב תוצאות טובות, אך נוכל לשפר את הביצועים או להאיץ את האימונים על ידי קצב למידה אופטימלי.

נוכל למצוא את קצב הלמידה האופטימלי באמצעות שלושת השיטות הבאות:

- Systematic approach – נרצה לקבוע את קצב הלמידה כך שיגרום ל Loss לרדת בצורה מהירה. לצורך כך, נבחן זאת על ידי ניסוי פשוט בו אנו מגדילים את קצב הלמידה בהדרגה לאחר כל mini-batch, ובוחנים את ה Loss המתקבל כפי שניתן לראות באיור 24. הגדלת קצב הלמידה יכולה להיות בצורה ליניארית או מעריכית.



איור 24: השפעת קצב הלמידה על קצב ירידת ה Loss

בקצב למידה נמוך, ה Loss אמנם יורד אך בקצב איטי מאוד. כשנגיע לקצב הלמידה האופטימלי, נוכל לראות ירידה מהירה (כלומר נגזרת הגרף באיור 24 הכי גדולה עבור קצב הלמידה האופטימלי) בפונקציית ה Loss. אולם אם נמשיך להגדיל את קצב הלמידה, ה Loss יתחיל לעלות וזאת משום

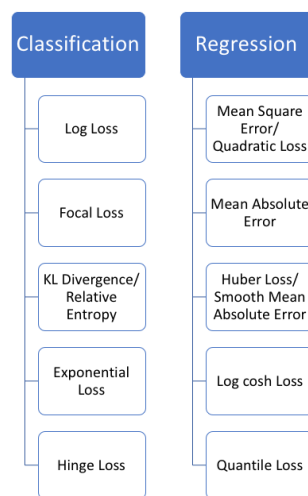
שעדכוני הפרמטרים בקצב למידה זה גורמים ל Loss לבצע "bounce around" ואף לסטות מהמינימום שנרצה להתכנס אליו.

- Schedule learning rate – בשיטה זו אנו מתחילים בקצב למידה גבוה יחסית ואז מורידים אותו בהדרגה במהלך האימון.
- Cyclical learning rate – בשיטה זו מגבילים את קצב הלמידה בתחום מסוים כך שקצב הלמידה משתנה ב – mini batch-ים שונים.

פונקציית שגיאה – Loss

בלמידה עמוקה המטרה הסופית היא מזעור או מקסום של פונקציית ה Loss. פונקציית ה Loss משמשת כמדידה עד כמה המודל יוכל לחזות את התוצאה הצפויה.

אין פונקציית Loss אחת שעובדת עבור כל סוגי הנתונים. את פונקציית ה Loss ניתן לסווג לשתי קבוצות: Classification and Regression Loss. Classification משמש לחיזוי לייבל הקלאס באופן בדיד בעוד שב - Regression המשימה לחזות מספר רציף.



איור 25: Classification and Regression Loss

חלוקת ה – dataset

את ה – dataset נחלק לשלושה:

- Training Dataset: המידע שבפועל אנו משתמשים כדי לאמן את המודל ובאמצעותו מעדכנים את המשקולות של רשת הניורונים.
- Validation Dataset: מידע זה משמש להעריך את המודל שאמנו, הערכה זו מתבצעת לאחר כל epoch או כמות מסוימת של batch-ים. אנו משתמשים במידע זה בכדי לכוון את ה – hyperparameter של המודל. נדגיש שהמודל רואה ומעביר את המידע הזה ברשת, אך לעולם אינו "לומד" ממנו (כלומר לא מעדכן את המשקולות). אנו משתמשים בתוצאות של הולידציה וכך יודעים לכוון את ה – hyperparameter בצורה נכונה יותר שתגרום לרשת להתכנס מהר יותר, ללמוד באופן מדויק ונכון ועוד.
- Test Dataset: זה המידע שנכנס לרשת לאחר שהיא אומנה לחלוטין. מערך המידע הזה עוזר לנו לדעת האם המודל שלנו יפעל כראוי גם על מידע שהוא טרם "ראה".

יחס חלוקת ה – Dataset

יחס חלוקת מערך המידע זה תלוי בעיקר בשני דברים. ראשית, גודל מערך המידע הכולל, שנית, על המודל בפועל שנרצה לאמן, יש מודלים שזקוקים למערך מידע גדול כדי להתאמן עליהם.

מודלים עם מעט hyperparameter, קלים יותר לקביעת אותם פרמטרים כך שנוכל להקטין את גודל ה - validation dataset, אך אם למודל יש hyperparameter רבים, נרצה validation dataset גדול יותר (אם כי כדאי לשקול גם אימות צולב). אולם בסופו של דבר, קביעת חלוקת מערך הנתונים תלוי מקרה.

לרוב משתמשים בחלוקה הבאה: נפצל את מערך המידע שלנו ל- 2 - Train and Test. לאחר מכן, שומרים בצד את ה- test dataset, ובחרים אחוז מסוים ממערך המידע שנשאר ל- training data ואת השאר ל- validation dataset כפי שניתן לראות באיור 26.



איור 26: פיצול ה - DataSet

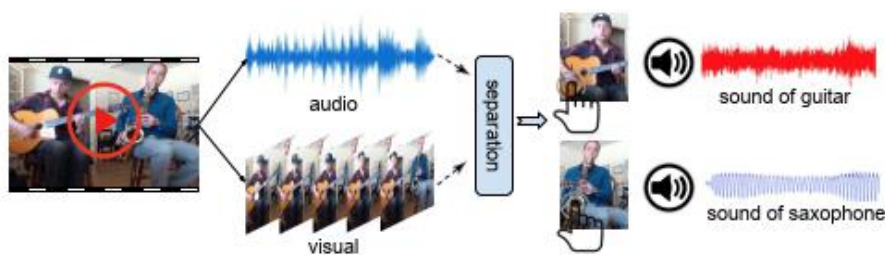
לאחר שהסברנו והרחבנו אודות תהליך הלמידה, מושגים, שיטות ועוד, נוכל כעת להרחיב על בעייתנו.

מבוא

הבנת סצנות ואירועים היא מטבעה חוויה רב-אופנית. אנו תופסים את העולם על ידי מבט והקשבה כאחד (ונגיעה, ריח וטעימה). חפצים מייצרים צלילים ייחודיים עקב תכונותיהם הגופניות והאינטראקציות שלהם עם עצמים אחרים והסביבה. לדוגמה, תפיסת סצנת חנות קפה עשויה לכלול ראיית כוסות, צלוחיות, אנשים ושולחנות, אך גם את שמיעת רעש הכלים, טחינת מכונת האספרסו והמנהל שמחלק פקודות לעובדיו. למידה התפתחותית אנושית היא מטבעה גם רב-אופנית, כך לדוגמה ילדים צעירים צוברים במהירות ראייה ויזואלית של חפצים והצלילים שלהם: כלבים נובחים, חתולים מייללים, טלפונים מצלצלים וכדומה.

עם זאת, בעוד שיש התקדמות משמעותית בזיהוי על ידי "הסתכלות" - גילוי אובייקטים, פעולות או אנשים על סמך המראה שלהם – השיטות לא תמיד עובדות בצורה טובה. למרות היסטוריה ארוכה של תיוג סרטוני audio-visual [17,18], רוב הפעמים מנותחים עצמים בווידיאו כאילו היו ישויות אילמות בסביבות שקטות. אתגר מרכזי הוא שבסרטון ריאליסטי, צלילי האובייקט נצפים לא כישויות נפרדות, אלא כערוך שמע יחיד שמערבב את כל התדרים שלהם יחד. הפרדת מקור שמע, למרות שנחקרה רבות בספרות עיבוד האודיו [19,20], נותרה בעיה קשה. קיימות שיטות שמתפקדות בצורה טובה על ידי לכידת הקלט באמצעות מיקרופונים מרובים, או מניחים שקבוצה נקייה של דוגמאות שמע מקוריות זמינה להשגחה (למשל הקלטה של כינור בלבד, הקלטה אחרת המכילה רק תוף וכו'), תנאים מוקדמים אלו מגבילים אותנו. משימת הפרדת השמע ה"עיוורת" מעוררת אתגרים הדומים ל- image segmentation ואולי יותר, מכיוון שכל הצלילים עולים אחד על השני באות הקלט.

המטרה שלנו היא ללמוד כיצד נשמעים אובייקטים שונים על ידי התבוננות והאזנה לסרטון לא מתויג המכיל חפצי שמע מרובים. אנו מציעים גישה weakly-supervised לפירוק שמע מעורב למקורות הצליל המרכיבים אותו. התבונה העיקרית היא שהתבוננות בצלילים במגוון ההקשרים החזותיים חושפת את הרמזים הדרושים להפרדת מקורות השמע; ההקשרים החזותיים השונים מעניקים פיקוח חלש לגילוי האסוציאציות. לדוגמה, לאחר שהתנסו בעבר בכלי נגינה שונים בשילוב שונה, ואז אנו מקבלים וידיאו עם גיטרה וסקסופון כמתואר באיור 27, אנו כבר יודעים באופן טבעי לאילו צלילים לצפות שיכולים להיות בשמע הנלווה, ולכן להפריד ביניהם בצורה טובה יותר. אכן, מדעני המוח מדווחים כי ה- MMN (mismatch negativity), נוצרת רק כאשר התבנית הוויזואלית מקדמת את ההפרדה של הצלילים [16]. זה מצביע על כך שמצג סינכרוני של גירויים חזותיים אמור לעזור לפתור את העמימות הצלילית בגלל מקורות מרובים, ולקדם תפיסה משולבת או מופרדת של הצלילים.



איור 27: הפרדת השמע לצלילים המרכיבים אותו

ומכאן אנו מציעים גישה חדשנית מהמאמר להפרדת מקור audio-visual המממש את האינטואיציה הזו. השיטה הראשונה משתמשת באוסף גדול של קטעי וידיאו שלא סומנו (כלומר לא מתויגים) כדי לייצור/לגלות ייצוג צלילי לכל אובייקט. באופן מדויק יותר, אנו משתמשים בכלי זיהוי תמונות עדכניים כדי לדעת אלו אובייקטים קיימים בכל וידיאו, ובמקביל אנו מבצעים NMF (שהוא פירוק המטריצה לכל של שני מטריצות ששואפות בקירוב למטריצה המקורית) על השמע של כל אחד מהווידיאו כדי לקבל את מערך וקטורי בסיס התדרים שלו. כפי שצינו, בשלב זה עוד לא ידוע איזה בסיסי שמע שייכים לאילו אובייקטים שנצפים בסרטון. כדי לעשות זאת, אנו בונים רשת נירונים שמממשת את רשת ה- multi instance multi label- MIML שתפקידה למפות את בסיסי השמע לאובייקטים חזותיים שגילינו בסרטון. מרשת זו, אנו מחלצים את בסיסי

השמע המקושרים לכל אובייקט חזותי, ומסיקים את דפוס הספקטרום שלהם. לבסוף, כאשר אנו מקבלים סרטון חדש, אנו משתמשים בבסיסי האודיו שלמדנו לכל אובייקט בכדי לבצע הפרדה של מקור האודיו בצורה טובה.

החידוש ברעיון שהפרייקט שלנו מתבסס עליו זה הוא שבניסיונות קודמים להפרדת מקור שמע בעזרת החלק הוויזואלי נתקלו בבעיה של מתאם קורלציה נמוך בין החלק הוויזואלי לחלק השמיעתי (אודיו) [21,22]. אנו מציעים ללמוד את תבנית הצליל ברמת האובייקט ממאות אלפי סרטוני וידיאו ללא תווית, וכך להכליל זאת להפרדת מופעים audio-visual חדשים.

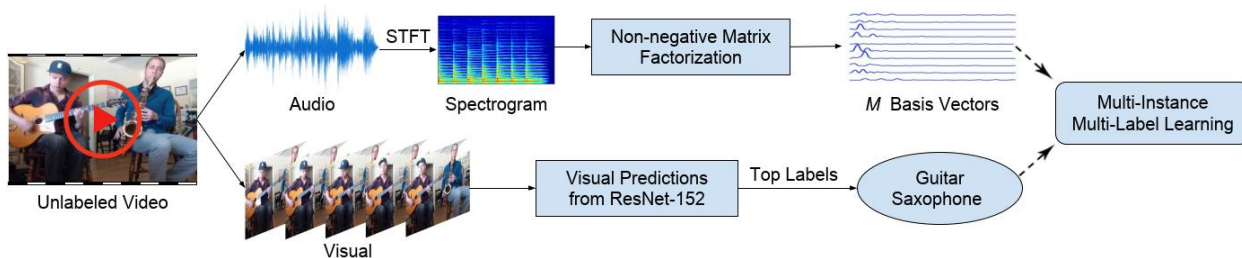
למרות חידוש המחקר על למידה שמשלבת מידע שמתקבל מיותר ממודליות אחת כלומר מתמונות ואודיו ועושה שימוש במידע ויזואלי – שמיעתי (משלב גם אודיו ותמונות) למשימות שונות [23,24], המחקר מתייחס לשמע כאל קלט מונוליטי יחיד (כמקשה אחת), ולכן הוא אינו יכול לשייך צלילים שונים לאובייקטים שונים באותו וידיאו. ולכן נעסוק בתרומות העיקריות והמשמעותיות בפרוייקט זה.

ראשית, נשפר את הפרדת מקור השמע בסרטונים על ידי "פיקוח" עליו, וזה בעצם יהיה המידע החזותי שנקבל מתוצאות זיהוי תמונות (דרך רשת Resnet-152). שנית, נבנה רשת למידה שנקראת – MIML – multi instance multi label שלומדת את הדפוסים הספקטריים של חפצים אקוסטיים שונים כך שנקבל ייצוג לצליל של כל אובייקט, ולאחר מכן, נכניס ייצוגים אלו למבנה הפרדת מקור השמע - NMF.

תקציר רצף המערכת

השיטה לומדת מה אובייקט משמיע מתוך סרטוני וידאו בעלי כמה מקורות שמע לא-מתויגים. בהינתן סרטון חדש, השיטה מחזירה את ערוצי השמע המופרדים ואת האובייקטים הנראים ששייכים אליהם.

1. הפרדת השמע ע"י חילוץ בסיסי השמע באמצעות NMF (בחלק "הוצאת וקטורי הבסיס").
2. זיהוי אובייקטי audio-visual הנמצאים בפריימים עבור סרטון לא מתויג באמצעות ה- ResNet152 (אובייקטים שמשמיעים צליל. כגון: כלי מנגינה, חיות וכו') מתוך הסרטונים הלא-מתויגים.
3. מעבירים את וקטורי הבסיס שקיבלנו ב-1 ואת הלייבלים שקיבלנו ב-2 לתוך הרשת MIML (Multi-instance multi-label)
4. נשתמש ברשת ה- MIML על מנת לקשר בין וקטורי בסיס עם אובייקטי ה- audio-visual.
5. הפרדת מקורות שמע עבור סרטוני וידאו באמצעות שיטת ה- semi-supervised NMF.



איור 28: רצף מערכת הלמידה

באיור 28 ניתן לראות את רצף מערכת הלמידה ה- Unsupervised (ללא-הנחיה. של אדם למשל). עבור כל וידאו:

בענף העליון, נבצע פירוק NMF על הספקטוגרמה של השמע שלו בכדי לקבל M וקטורי בסיס. בענף התחתון, נפעיל רשת ResNet-152 (שכבר אומנה על ImageNet) על מנת לקבל חיזויים (Labels) על אובייקטים ה- audio-visual שמופיעים בסרטון. לבסוף, מכניסים את מוצא שני הענפים לעיל לרשת ה- MIML כדי להחליט אלו וקטורי בסיס שייכים לאיזה אובייקט שזיהינו קודם.

הוצאת וקטורי הבסיס

הפרדת ערוץ אחד של מקור שמע היא בעיה של השגת שערור - s_j עבור כל אחד מ- J מקורות השמע,

$$x(t) = \sum_{j=1}^J s_j(t) \quad \text{כאשר } s_j(t) \text{ הן אותות בזמן-בדיד.}$$

התמרת STFT

נמיר את האות הכולל - x לספקטרום שלו - $V \in \mathbb{R}_+^{F \times N}$ שמכילה F תדרי בסיס, ו N מסגרות של STFT (Short-time Fourier transform), כאשר התמרת STFT מתוארת באיור 29. התמרה זו מראה

את השינוי של התדר והפאזה של האות בזמן, כלומר עבור כל אינטרוול זמן אנו נראה את התדרים שהיו בזמן זה. אנו נפעל בתחום התדר, ולבסוף נחזיר חזרה לזמן באמצעות (Inverse Short-time) STFT (Fourier transform).

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \sum_{m=-\infty}^{\infty} X_m(\omega) e^{j\omega n} d\omega$$

$$\text{STFT}\{x[n]\}(m, \omega) \equiv X(m, \omega) = \sum_{n=-\infty}^{\infty} x[n] w[n-m] e^{-j\omega n}$$

איור 29: עליון: תיאור מתמטי להתמרת STFT, תחתון: STFT

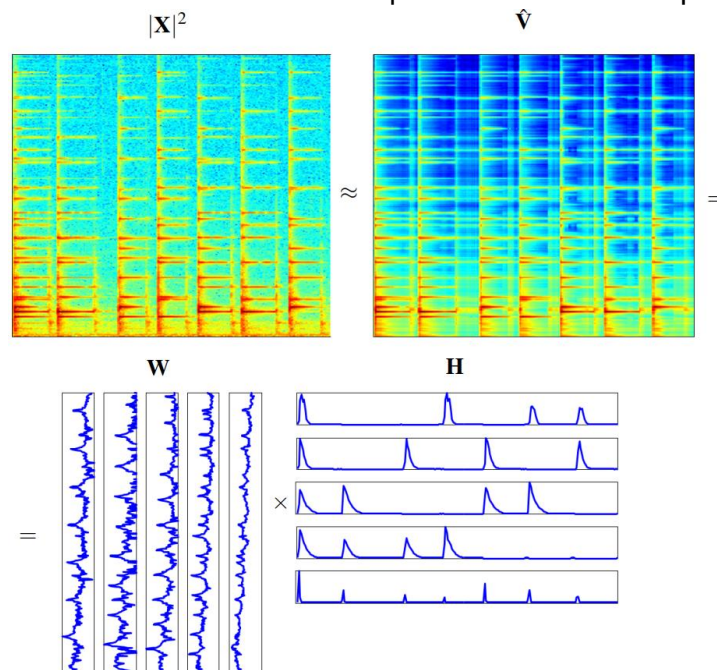
פירוק NMF

NMF - Non-negative matrix factorization, זוהי שיטה לפיצול מטריצה חיובית V (במקרה שלנו $V=|\text{STFT}(\text{audio})|$) לכפל מטריצות $V \approx \tilde{V} = WH$ כאשר $H \in R_+^{M \times N}$ ו- $W \in R_+^{F \times M}$. במקרה שלנו העמודות של מטריצת W מייצגות את **וקטורי הבסיס** (אשר שייכים למקורות שמע ספציפיים) ומטריצת H שמייצגת את ההגבר שלהן. על מנת למצוא את המטריצות המתאימות, נעשה שימוש באופטימיזציה של המשוואה $\min_{W,H} D(V|WH)$ כאשר D זהו חישוב ה- *divergence* שמבצעים ע"פ *KL divergence*. כאשר בכל איטרציה אנו מעדכנים את המטריצות ע"פ האלגוריתם הבאה-

$$W_{[i,j]}^{n+1} \leftarrow W_{[i,j]}^n = \frac{\left(V (H^{n+1})^T \right)_{[i,j]}}{\left(W^n H^{n+1} (H^{n+1})^T \right)_{[i,j]}} \quad H_{[i,j]}^{n+1} \leftarrow H_{[i,j]}^n = \frac{\left((W^n)^T V \right)_{[i,j]}}{\left((W^n)^T W^n H^n \right)_{[i,j]}}$$

*הכפלה של המטריצות כאן מתבצעות איבר - איבר.

להמחשת הרעיון, להלן דוגמה שמציגה את הפירוק הנ"ל:



איור 30: NMF שבוצע על הספקטרום של רצף קצר של פסנתר שמורכב מחמישה תווים

כך שבסוף שלב זה נקבל את M וקטורי הבסיס הרצויים.

מערכת למידה לזיהוי אובייקטי-קול – Weakly-Supervised

יכולים להופיע בסרטון כמה אובייקטים בו-זמנית ובאופן דומה בשמע המצורף. בנקודה זו, עדיין לא ידוע מי מוקטורי הבסיס שחילצנו קודם(עמודות המטריצה W) שייכים לאיזה מהאובייקטים שבוידאו. בכדי לגלות את שייכות זו, אנו תכננו מערכת למידה – MIML שתתאם בין וקטורי הבסיס לאובייקטים שזיהינו.

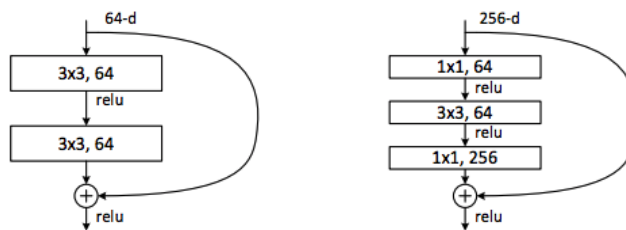
כפי שצינו בהקשר לאיור 28, בהינתן סרטון לא-מתויג, אנו מחלצים ממנו את הפריימים ואת השמע, ואז אנו מבצעים פירוק NMF על הספקטרום של השמע ומקבלים M וקטורי בסיס עבור כל וידאו.

רשת ה- ResNet

עבור הפריימים, אנו משתמשים ברשת ResNet-152 שאומנה על ImageNet (dataset שמכיל כמיליון תמונות)

החסרון ברשתות רגילות שרשת ה- ResNet באה לפתור הוא, שבכל רשת אנו מחשבים את גראדינט השגיאה בסוף הרשת ומשתמשים ב- backpropagation כדי לחלחל/להעביר הלאה את גראדינט השגיאה שלנו לאחור(לכיוון תחילת הרשת כפי שהסברנו בחלק מבוא ללמידה עמוקה) דרך הרשת. על פי כלל השרשרת, עלינו להמשיך להכפיל את הביטויים עם גראדינט השגיאה כל זמן שאנו הולכים אחורה לתחילת הרשת. אולם, ברשת עמוקה מאוד, אם נכפיל ביטויים רבים שהם פחות מ-1, התוצאה הסופית תהיה קטנה מאוד. לכן, הגראדינט יהיה קטן מאוד ככל שאנו מתקרבים לשכבות הראשונות בארכיטקטורה עמוקה. גראדינט קטן יהיה בעייתי מכיוון שאז איננו יכולים לעדכן את פרמטרי הרשת בצורה יעילה מספיק והלמידה תהיה איטית מאוד. במקרים מסוימים נקבל שהגראדינט יהיה אפס, כלומר איננו מעדכנים את הפרמטרים הקודמים כלל(זוהי הבעיה שהצגנו בחלק מבוא ללמידה עמוקה – vanishing gradient).

כדי להתגבר על בעיה זו, אנו משתמשים בנוסף, בפונקציית הזהות כדי לשמר את הגראדינט. ניתן לראות את הסכמה של בלוק בודד ברשת ה- ResNet באיור 31 ואת התיאור מתמטי שגורם להוספת איבר בגראדינט ולא רק הכפלות של ביטויים שיכולים לשאוף לאפס, באיור 32.



איור 31:בלוק בודד ברשת ResNet

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial H} \frac{\partial H}{\partial x} = \frac{\partial L}{\partial H} \left(\frac{\partial F}{\partial x} + 1 \right) = \frac{\partial L}{\partial H} \frac{\partial F}{\partial x} + \frac{\partial L}{\partial H}$$

add_gradient

איור 32: תיאור מתמטי להוספת הגראדינט ברשת ה- ResNet

כאשר L היא פונקציית ה- Loss x זהו הכניסה, F אלו השכבות H זהו מוצא בלוק ה- F המסתכם עם איבר הכניסה x.

רשת ה- ResNet נועדה לזהות לאיזו קטגוריה האובייקטים שבפריימים שייכים(כלומר מה הם). לכן, נכניס אל רשת זו את עשרת הפריימים שמתאימים לסרטון מסוים ולאחר מכן נבצע max-pool על מוצאי הרשת של פריימים אלו. הלייבלים העליונים (עם הסתברות גבוהה מסוף מסוים) משמים כלייבלים "חלשים" עבור הוידאו.

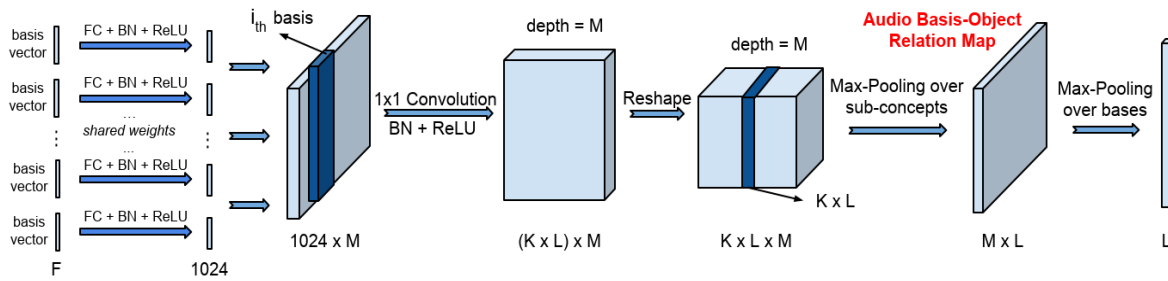
* לייבלים חלשים – הסרטון עצמו לא מתויג ע"י אדם למשל. אלא אנו משתמשים ברשת אחרת כדי לתייג את אותם פריימים.

לאחר ביצוע שני השלבים המתוארים לעיל – NMF and ResNet, נקבל את וקטורי הבסיס שחולצו עבור כל סרטון והחיזויים של האובייקטים שזוהו בו. שניהם מוזנים לתוך מערכת הלמידה – MIML בכדי למצוא שייכות ביניהם.

Deep MIML(Multi-Instance Multi-Label) Network

בחלק זה נרחיב ונפרט אודות מבנה רשת ה – MIML, כאשר הכניסות לרשת, אלו הווקטורי בסיס שחילצנו באמצעות ה – NMF והלייבלים החלשים באמצעות רשת ה – ResNet. מטרת רשת זו היא לקשר בין הווקטורי בסיס לאובייקטי ה- audio-visual המופיעים בסרטון, כך שבסופו של דבר נקבל עבור כל אובייקט את וקטורי הבסיס שמתאימים לו.

לצורך משימה זו, תכננו את ה – Deep MIML Network. רשת זו מקבלת קבוצה של וקטורי בסיס $\{B\}$ ככניסה לרשת, ועבור כל קבוצה יש M וקטורי בסיס $B_i, i \in [1, M]$ שחולצו מסרטון אחד. הלייבלים החלשים שאנו משיגים מהחיזוי של רשת ה ResNet-152 הן ברמת הקבוצה ולא עבור כל אחד מווקטורי הבסיס.



איור 33: מבנה רשת ה – MIML

באיור 33 ניתן לראות את מבנה רשת ה – MIML אשר לוקחת קבוצה של M וקטורי בסיס עבור כל וידאו ככניסה, ונותנת חיזויים ברמת הקבוצה של האובייקטים שנמצאים בשמע. החיזוי מרשת ה- ResNet שאומנה מ- ImageNet משמים ללייבלים "חלשים" בכדי לאמן את רשת זו עם סרטון לא-מתויג(ללא ידע מה מופיע בו). וקטורי הבסיס מוזנים דרך Siamese Network (רשת נזירונים שמשמשת במשקלים זהים בעוד שהיא מכניסה במקביל כמה כניסות ומחשבת את המוצאים) כאשר רשת זו בעלת M ענפים עם משקלים זהים. רשת ה – Siamese נועדה לצמצם את הממד של בסיסי תדרי השמע וללמוד את דפוסי ספקטרום השמע, ע"י שכבת FC+BN+ReLU. במוצא רשת ה- Siamese נקבל שכבה בעלת ממדים של $1024 \times M$, כאשר כל חתך שבה מייצג וקטור בסיס עם ממד מצומצם.

לאחר מכן, נרצה להביא את זה לצורה שעבור כל לייבל (מתוך L לייבלים) ישנם K תתי-תכנים ששייכים לו על מנת לתפוס גם את האלה עם משמעויות סמנטיות נסתרות, לדוגמה, עבור תופים, המשמעות הסמנטית הנסתרת יכולה להיות סוגים שונים של תופים כמו תוף בונגו, טאבלה וכו'.

ולכן, נרצה כעת להעביר את מוצא רשת ה – Siamese לתוך סט השכבות 1×1 Convolution-BN-ReLU שגורם להורדת ממד מ- $1024 \times M$ ל- $(K \times L) \times M$, ולאחר מכן, נבצע reshape כך שנקבל קובייה בממדים של $K \times L \times M$ (קודם זה היה דו-ממד), כאשר K זה מספר תתי-התכנים, L זה מספר קטגוריות האובייקטים ו- M זה מספר וקטורי בסיסי השמע.

המשמעות עבור גדלים אלה היא שבצעם עומק הטנסור ("קובייה") שווה למספר וקטורי הבסיס שנכנסו, כשכל חתך $K \times L$ תואם לבסיס מסוים אחד. כלומר ה activation score (רמת הפעלת הצומת ברשת) עבור צומת ה- $(k, l, m)_i$ בקובייה, מייצגת את הניקוד התואם עבור תת-התוכן ה- k_i של הלייבל ה- l_i עבור וקטור הבסיס ה- m_i .

בהמשך, על מנת לקבל חיזוי ברמת הקבוצה (כלומר עבור כל קבוצת הווקטורים), עשינו שתי שכבות של max-pooling, הראשון פועל על ממד התתי - תוכן (K) בכדי לייצר מפת יחס בין וקטורי הבסיס לאובייקטים. וה- max-pooling השני פועל על ממד וקטורי הבסיס (M) בכדי לייצר חיזוי ברמת הסרטון.

כדי לאמן רשת זו השתמשנו בפונקציית ה-Loss: Multi-Label Hinge Loss:

ב-Loss זה אנו דואגים שיהיה מרחק של אחד בין הלייבל הנכון ללא נכונים וכך עבור כל הלייבלים הנכונים - משימת multi label. כלומר במידה ווקטור המוצא שווה ל-x ואינדקסי ווקטור ה-target שווה ל-y אזי חישוב ה-Loss יתבצע באופן הבא - וזאת בהתאם להגדרת ה-Loss כפי שניתן לראות באיור 34.

$$\text{loss}(x, y) = \sum_{ij} \frac{\max(0, 1 - (x[y[j]] - x[i]))}{x.\text{size}(0)}$$

where $x \in \{0, \dots, x.\text{size}(0) - 1\}, y \in \{0, \dots, y.\text{size}(0) - 1\}, 0 \leq y[j] \leq x.\text{size}(0) - 1$, and $i \neq y[j]$ for all i and j .

איור 34: Hinge Loss כאשר x הוא חיזוי הרשת ו-y וקטור מאפס עד גודלו של x

שכפי שהסברנו קודם פונקציית ה-Loss הנ"ל מעודדת את ה-score של החיזויים של ה-class-ים הנכונים, להיות גדולים יותר מ-האל נכונים ב-1.

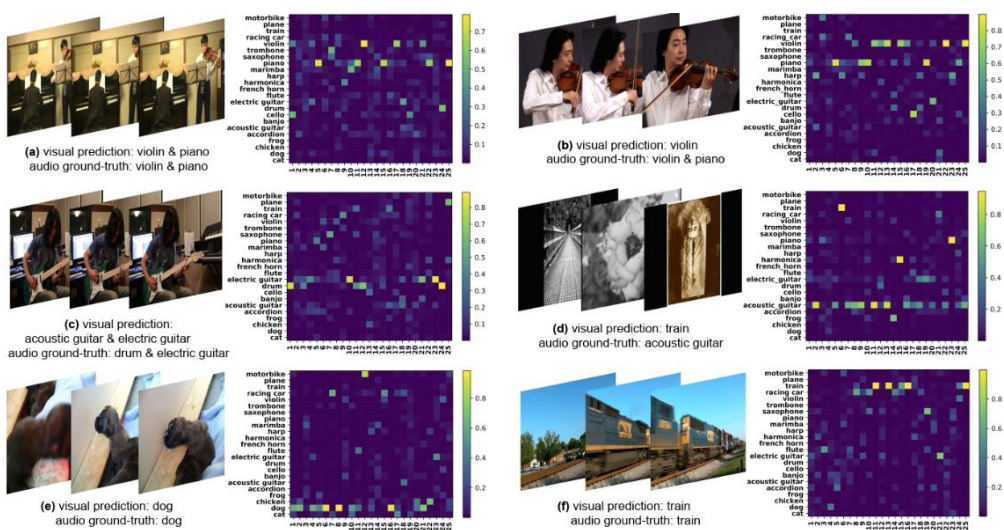
הפרדת וקטורי-בסיס לכל אובייקט

רשת ה-MIML, לומדת את הקשרים בין השמע לחלק החזותי, אך אינה מפרידה בין הקשרים הללו. הקולות שבשמע והאובייקטים שנמצאים בפריימים של הסרטונים הלא-מתוגים הם מגוונים ורועשים. וקטורי הבסיס שחילצנו מכל וידאו יכולים להיות מרכיבים שמשותפים לכמה אובייקטים או אפילו בכלל לא משויכים לאובייקטים שזוהו. החיזוי-מראה מרשת ה-ResNet-152 הינם חיזויים משוערים על האובייקטים שעלולים להיות, אך הם בהחלט לא תמיד אמינים.

ולכן, על מנת להשיג וקטורי בסיס איכותיים נבחרים עבור כל קטגוריה של האובייקטים, נשתמש ברשת ה-deep MIML המאומנת בתור כלי. מפת היחס בסיסי שמע - אובייקט שלאחר שכבת ה-Pooling הראשונה ברשת ה-MIML, מייצרת score תואם בכל וקטורי הבסיס עבור כל הלייבלים של האובייקטים. נפעיל את פונקציית ה-softmax על כל אחד מממד הבסיסים (M) בכדי לנרמל scores שתואמים לאובייקטים להסתברויות.

בהסתכלות על המפה המנורמלת, אנו יכולים לגלות את הקישורים מהבסיסים לאובייקטים. אנחנו לקחנו רק את הבסיסים שנתנו הסתברות גבוהה לאובייקטים הנכונים (האובייקטים שזוהו קודם).

ניתן לראות באיור 35 כמה דוגמאות של מפות יחס בסיס-אובייקט כאשר בכל דוגמה, אנו מראים את הפריימים, החיזוי של האובייקטים ואת מפת היחס הבסיס-לייבל שחזתה ע"י רשת ה-MIML.



איור 35: דוגמאות של object relation map של basis - object

הפרדת אובייקט-קול עבור סרטון חדש

בשלב זה, אנו מציגים את התהליך שנועד להפריד מקורות שמע בסרטונים חדשים. כפי שניתן לראות באיור 36, בהינתן סרטון בחינה חדש, q :

1. אנו מחשבים את מגניטודת הספקטוגרם של השמע שלו - V^q באמצעות STFT
2. מזהים אובייקטים ע"י שימוש ברשת ה- ResNet
3. מבצעים NMF על השמע בשביל לקבל את וקטורי הבסיס שאותם נכניס ל- MIML
4. ממוצא רשת ה- MIML נבחר את וקטורי הבסיס המתאימים לאובייקטים הנכונים ושעוברים סף מסוים(כפי שמתואר בחלק קודם), כך נקבל שלכל אובייקט נכון(אובייקט שקבלנו שהוא צריך להיות בסרטון ע"פ הנחיית ה- ResNet) יהיו מספר וקטורי בסיס המתאימים עבורו.
5. נשתמש בוקטורי הבסיס שמצאנו בסעיף קודם בכדי להנחות את הפרדת מקורות השמע שלנו שמבוססת על פירוק NMF. בצורה מתמטית יותר:

$$V^q \approx \tilde{V}^q = W^q H^q$$

$$= \begin{bmatrix} W_1^q & \dots & W_j^q & \dots & W_J^q \end{bmatrix} \begin{bmatrix} H_1^q & \dots & H_j^q & \dots & H_J^q \end{bmatrix}^T$$

כאשר J מייצג את מספר האובייקטים שזוהו (J מקורות שמע פוטנציאליים), ו- W_j^q מכיל את הבסיסים ששלפנו (ממה שלמדנו קודם) שתואמים לאובייקט ה- j בסרטון הכניסה - q .

במילים אחרות, אנו משרשרים את וקטורי הבסיס עבור כל אחד מהאובייקטים שזוהו(כפי שמתואר בסעיף 4) כדי לבנות את מטריצת הבסיסים - W^q . לאחר מכן, אנו מקבעים

את מטריצת וקטורי הבסיסים ורק משערכים את רמות ההפעלה שבמטריצה H^q עם כללי ה- multiplicative update: בעיית ה- NMF אינה קמורה ב- W ו- H אך היא קמורה רק ב- W או רק ב- H . לכן, נוכל לפתור את הבעיה, ע"י ביצוע אופטימיזציה ביחס ל- H תוך שמירה על W קבוע שמצאנו:

$$\min_{H \in \mathbb{R}^{k \times m}} \|V - WH\|_F^2 \text{ s.t. } W, H \geq 0 \Rightarrow H \leftarrow H - \eta_H \cdot \nabla_H f(W, H) \Rightarrow$$

$$H \leftarrow H \cdot \frac{W^T V}{W^T W H^T}$$

6. משיגים את הספקטרום שתואם לכל אובייקט שזוהו באמצעות החישוב -

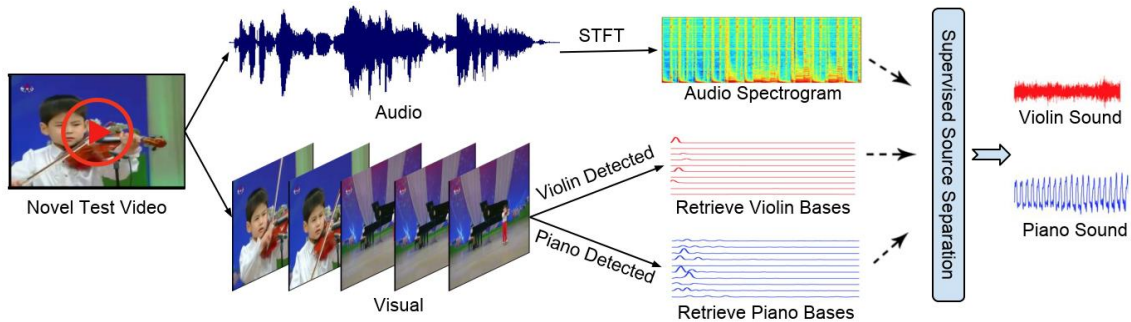
$$V_j^q = W_j^q H_j^q$$

7. נשחרז אותות ממקור שמע ספציפי ע"י הפעלת softmax על הספקטוגרם הכולל:

$$\mathbb{V}_j = \frac{V_j^q}{\sum_{i=1}^J V_i^q} \mathbb{V}$$

כאשר \mathbb{V} מכיל גם את המגניטודה וגם את הפאזה - התמרת STFT של השמע.

8. לבסוף, אנו מבצעים ISTFT על \mathbb{V}_j בכדי לשחזר את אותות השמע עבור כל אחד מהאובייקטים שזוהו. אם אובייקט שזוהו אינו משמיע צליל אז ה- *activation scores* שלו יהיו נמוכים. חלק זה יכול להיחשב כצורת *self-supervised NMF*, כאשר האובייקטים שזוהו מגלים איזה בסיסים רלוונטיים להנחות הפרדת שמע.



איור 36: תרשים זרימה לשלב ה- test

יישום השיטה

קעת אנו מאמתים את שיטתנו ומשווים אותה לשיטות קיימות (כפי שהצגנו במבוא)

Datasets

הדאטא שאנו עוסקים הוא מ- AudioSet אשר איננו מתויג

AudioSet-Unlabeled

אנו משתמשים ב-AudioSet כמקור לסרטונים לא מתויגים שנאמן בעזרתם את הרשת. מערך המידע מורכב מרשימה של כתובות אתרים ל YouTube שבהם מקטעי וידיאו קצרים של 10 שניות המתרכזים לעתים קרובות באירוע אחד. עם זאת, השיטה אינה מניחה הנחות מסוימות לגבי השימוש בסרטונים קצרים או בקטעי וידיאו, וזאת מכיוון שהיא לומדת את הבסיסים (הייצוג לצליל מסוים) בתחום התדר ומשלבת בתוכה הן את החיזוי הוויזואלי והן את בסיסי השמע מכל המסגרות (frames).

סינון הדאטא

למרות האמור, הסרטונים מאתגרים וזאת מכיוון שרבים באיכות ירודה וישנם רעשים שאינם קשורים לצלילי האובייקט, כגון גלי סינוס, הד, אינפרה-סאונד (שמע בתדר נמוך) וכו'. לכן לאחר שהורדנו את הדאטא אנו מסננים את הדאטא לאלו שעלולים להציג אירועי audio-visual, כולל הקטגוריות הבאות: אקורדיון; גיטרה אקוסטית; בנג'ו; צ'לו; תוף; גיטרה חשמלית; חליל; קרן יער; מפוחית; נבל; מרימבה; פסנתר; סקסופון; טרומבון; כינור; כלב; חתול; צפרדע; עוף, תרנגול; אוטו; אופנוע; הובלת רכבות; מטוסים. בעזרת הפיצול שמערך ה- AudioSet זה מספק לנו אנו מיעדים באופן רנדומלי חלק מקטעי הוידאו כ- validation data ואת השאר בחלק זה כ- training data וחלקו השני של מערך נתונים זה כ- test data.

לאחר הסינון המוזכר לעיל קיבלנו שהנתונים הסופיים של מערך הנתונים AudioSet-Unlabeled מכילים עבור:

- Train – 180,477 דוגמאות.
- Val – 7,520 דוגמאות.
- Test – 2,329 דוגמאות.

את המימוש לסינון המוזכר לעיל ניתן לראות בפונקציית `subsetOfClassesAll` באיור 37. הפונקציה מסננת את האובייקטים שאינם audio-visual ע"י לקיחת האינדקסים המתאימים המתקבלים מרשת ה-ResNet. בנוסף לכך, אנו מכווצים אינדקסים אשר מסמלים תת קטגוריות לקטגוריה מסוימת לידי אינדקס אחד, כך למשל, עבור סוגים שונים של תופים נכווץ את כולם לכדי אינדקס אחד כך שיסמל את התופים באופן כללי. פעולה זו של כיווץ אינדקסים ולקיחת האובייקטים שהם audio-visual בלבד מסייעת לנו לרדת מ-1000 קלאסים שיש ב-ResNet ל-23 קלאסים.

```
def subsetOfClassesAll(label):
    selected_label = np.zeros(23)
    #4 animals: cat, dog, chicken, frog
    indexes = [[281,285],[151,275],[7,8],[30,32]]
    for i,indexlist in enumerate(indexes):
        start = indexlist[0]
        end = indexlist[1]
        for j in range(start, end+1):
            selected_label[i] = selected_label[i] + label[j]
        selected_label[i] = selected_label[i] / (end - start + 1)
    #15 music instruments: accordion, acoustic_guitar, banjo, cello, drum, electric_guitar, flute, french_horn, harmonica, harp,
    marimba, piano, saxophone, trombone, violin
    indexes = [[401], [402], [420], [486], [541], [546], [558], [566], [593], [594], [642], [579,881], [776], [875], [889]]
    for i,class_indexes in enumerate(indexes):
        for index in class_indexes:
            selected_label[i+4] = selected_label[i+4] + label[index]
        selected_label[i+4] = selected_label[i+4] / len(class_indexes)
    #4 vehicles: racing_car, train, plane, motor_scooter
    indexes = [[751,817,511], [705,466,820,547], [726,404,895], [670,665]]
    for i,class_indexes in enumerate(indexes):
        for index in class_indexes:
            selected_label[i+19] = selected_label[i+19] + label[index]
        selected_label[i+19] = selected_label[i+19] / len(class_indexes)
    return selected_label
```

איור 37: סינון מערך הנתונים

Implementation Details

בחלק זה נסביר את המימוש בפועל שעשינו.

הכנת הדאטא

לאחר שקיבלנו את הדאטא המסונן –

1. הורדנו וערכנו את הסרטונים באמצעות הספריית `pafy`, ו-`ffmpeg`. אשר הראשונה מורידה את הסרטון מ-YouTube והשנייה מחלצת מהסרטון את השמע ואת הפריימים (10 פריימים, 1 לכל שנייה).
2. דגמנו מחדש את האודיו בתדר של $f_s=48\text{kHz}$, ביצענו עליו STFT בגודל 2401×201 כאשר:
 - 2401 אלו מספר התדרים המתקבלים עבור אורך חלון $L_h=0.1\text{s}$ המניב $4,800$ דגימות, ולכן מספר התדרים המתקבלים הם $4800/2+1=2401$ (הדגימות ממשיות ולכן יש סימטריה בהתמרה ולכן אפשר לקחת רק חצי מהתדרים+תדר אפס)
 - 201 הם מספר החוצצים/מסגרות שמבצעים עליהם את ההתמרת STFT המתקבלים כתוצאה מהאורך החלון שציינו לעיל וחפיפה של חצי אורך החלון, $R=L_h/2$.
3. נבצע פירוק NMF על ההתמרה המתקבלת באמצעות התכנסות Kullback – Leibler ועדכון של `multiplicative update` שהצגנו בחלק "הפרדת אובייקט קול עבור סרטון חדש". ובכך, נחלק $M = 25$ וקטורי בסיס מכל שמע.
4. שינוי גודל הפריימים משתנה לגודל הבא: 224×224 .
5. חילוף הלייבלים שמופיעים בפריימים באמצעות הרשת ResNet152
6. את וקטורי הבסיס שהתקבלו מה- NMF והלייבלים המתקבלים עבור הפריימים המתאימים מרשת ResNet נשמור בקובץ שנוכל לקרוא בשלב אימון הרשת.

בניית הרשת

בתחילה ניסינו לבנות את הרשת על תשתית של Keras עם backend של TensorFlow. אך כשהגענו לשלב של בחירת ה Loss, זיהינו שאין Loss עם פונקציונליות זהה לאותו ה Loss שהשתמשו במאמר(שבוצע על ספריית Pytorch). לכן, נאלצנו להשתמש ב Loss אחר.

ניסינו תחילה להשתמש ב Hinge Loss(רעיון דומה למה שיש ב Pytorch אך הפונקציונליות שונה), בשילוב עם שכבת אקטיבציה Sigmoid/Softmax בסוף הרשת(וגם בלי). אך למערכת היה קשה להגיע למינימום ה Loss האופטימלי, והיא די מהר נתקעה במינימום מקומי בערכי ה Loss ולא השתפרה מאז. אז ניסינו לשנות את ה Learning rate לערכים גבוהים יותר או קטנים יותר, וכן גם את קצב דעיכתו. אך עדיין התוצאות לא השתפרו.

אותו דבר ניסינו עם Binary Cross Entropy Loss אך עדיין אותה הבעיה.

ולכן, ניסינו לבנות את ה Loss בעצמנו עם אותה הפונקציונליות של Hinge שציינו. כיוון ש Keras דורש להשתמש בפעולות backend בעת בניית ה Loss היינו די מוגבלים. אך בכל זאת הצלחנו לבנות את ה Loss המתאים בצורה מאוד חסכונית ופשוטה (למעט השימוש ב max מה שדרש מאיתנו להשתמש בשכבת sigmoid). אבל עדיין התוצאות לא היו טובות והמערכת לא הצליחה להתאמן.

לבסוף, לאחר המאמצים הרבים נאלצנו לעבור לספריית PyTorch, ולבנות מחדש את הרשת שלנו. כאשר שם, כפי שציינו, קיים ה Loss המתאים עם הפונקציונליות הנכונה(MultiLabelMarginLoss). ואכן הפעם היא באמת הצליחה ללמוד ולהשתפר.

כפי שניתן לראות באיור 38 בנינו את הרשת בשכבות המתאימות שפירטנו בחלק "MIML", כאשר השתמשנו ב - MultiLabelMarginLoss ובאלגוריתם אופטימיזציה Adam שפירטנו אודותיו במבוא.

```
class MIML(nn.Module):
    def __init__(self):
        super(MIML, self).__init__()
        self.conv1 = nn.Conv2d(2401, 1024, 1) # Fx1xM to 1024x1xM
        self.bn1 = nn.BatchNorm2d(1024)
        self.conv2 = nn.Conv2d(1024, K*L, 1) # 1024x1xM to K*Lx1xM
        self.bn2 = nn.BatchNorm2d(K*L)
        self.pool1 = nn.MaxPool2d((K, 1), stride=(1,1))
        self.pool2 = nn.MaxPool1d(M)

    def forward(self, x):
        x = x.transpose(-1,-2) # bxFxM
        x = x.unsqueeze(dim=-2) # bxFx1xM
        x = nn.functional.relu(self.bn1(self.conv1(x))) # bx1024x1xM
        x = nn.functional.relu(self.bn2(self.conv2(x))) # bxK*Lx1xM
        x = torch.squeeze(x, dim=2) # bxK*LxM
        x = x.view(-1, L, K, M) # bxLxKxM
        x = self.pool1(x) # bxLx1xM
        miml_map = x.squeeze(dim=2) # bxLxM
        x = self.pool2(miml_map).squeeze(dim=2) # bxL
        return x, miml_map
```

איור 38: מבנה רשת ה - MIML

קצב הלמידה עבור אופטימיזציית Adam נקבעת ע"פ פונקציית scheduler שהגדרנו שבו נתחיל עם $lr=0.001$ ועבור כל איפוק חמישי נוריד אותו ב- 6%.

בניית Custom Data Generator

לצורך אימון הרשת נבנה את ה- Custom DataSet שתכניס לרשת את הדאטא עבור כל batch. במחלקה זו מימשנו את המתודות:

- `__init__`: מאתחלת את אתה - id של כלל הדאטא.
- `__len__`: מחזירה את גודל מערך הדאטא, מספר ה- ids בסה"כ.
- `__getitem__`: מקבלת את אינדקס הדגימה ב- batch ומחזירה עבורו את וקטורי הבסיס והלייבלים המתאים לאותה דגימה.

תוצאות

Train and Val Loss

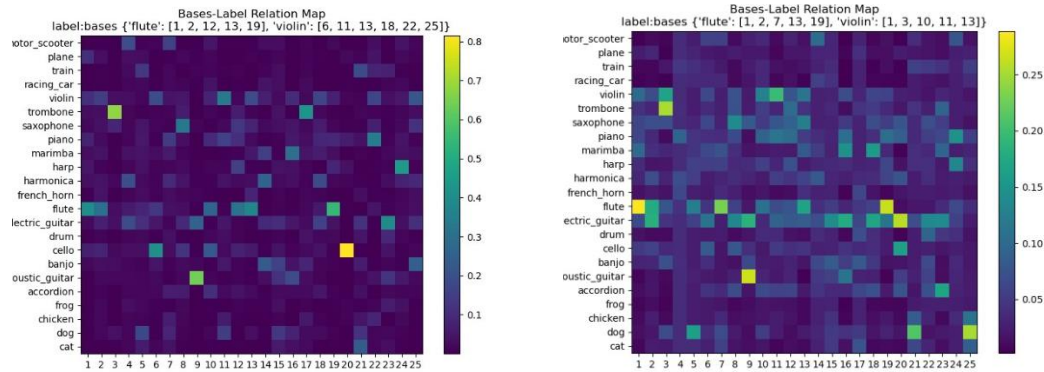
לאחר שאימנו את הרשת קיבלנו את גרף ה-Loss הבא:



איור 39: Loss של האימון והולידציה

כפי שניתן לראות מהאיור לעיל, בתחילת האימון הרשת אכן לומדת - Loss הולידציה והאימון יורדים, אולם לאחר מס' איפוקים רק Loss האימון ממשיך לעומת Loss הולידציה שמתחיל לעלות, כלומר אנו מזהים שמדובר ב- overfitting זאת משום שהדאטא שאיתו אנו מאמנים את הרשת הוא לא גדול מספיק ולכן הרשת "מתאימה" את עצמה בקלות לדאטא של האימון (כפי שהסברנו בהרחבה אודות ה- overfitting במבוא).

לכן מומלץ לקחת את המשקלים של הרשת באיפוק די קטן, עוד באזור ש ה val loss די נמוך, באזור איפוק 20 למשל. ואכן עשינו זאת, אבל אז גילינו שהמפות שיוצאות הן די רועשות(כפי שניתן לראות באיור 40), מה שמשפיע לבסוף על התוצאה הסופית(השמע עצמו). לכן ניסינו רבות עוד קומבינציות של משקלים ודוגמאות, עד שהגענו לתוצאות די טובות יחסית במשקלים של איפוק 140. שבה הרשת כבר מספיקה "חדה" במובנים של מה יש בשמע, והיא הרבה פחות רועשת.



איור 40: ימין – מפה מאיפוק 20. משמאל – מפה מאיפוק 140

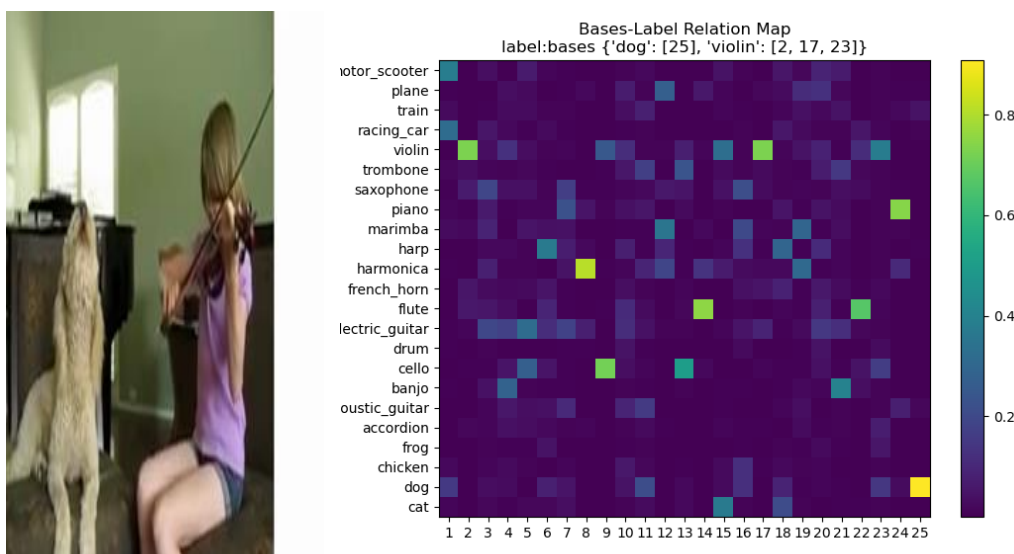
לכן, בחרנו ושמרנו את פרמטרי הרשת באיפוק 140.

מפות יחס בסיס – אובייקט

בחלק זה אנו נציג את התוצאות בכדי להמחיש את האפקטיביות של אימון רשת ה – MIML ואת ההצלחה בהפרדת מקור שמע. במטרה לבחון את הרשת אנו מבצעים בדיקה עם סרטונים בעלי מקורות שמע מרובים ממערך הדאטא – AudioSet.




כלב וכינור

באיור 41-43 ניתן לראות את הפריימים (נציג רק פריים בודד) שמסמלים את הלייבלים הנכונים ואת מפת יחס בסיס – אובייקט שבה לעומת זאת ניתן לראות בכותרת את הלייבלים שרשת ה – ResNet חזתה ואת האקטיבציות המתאימות לוקטורי הבסיס עבור כל לייבל.



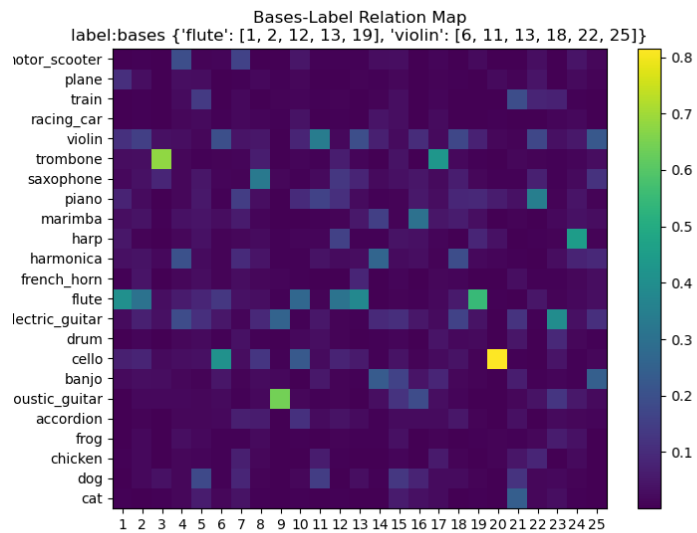
איור 41: ימין – מפת יחס בסיס – אובייקט, שמאל – פריים מהסרטון

הרשת אכן למדה את צלילי הכלב והכינור ולכן היא הפעילה את אקטיביות יחסית גבוהות עבור לייבלים אלו, בשביל לוודא שאכן ההפרדה בוצעה כראוי, נשמע את הפלט המתקבל עבור כל אחד מהמקורות שמע:

- 
 sound_original.wav • שמע מקורי:
- 
 sound_dog.wav • שמע הכלב:
- 
 sound_violin.wav • שמע הכינור:




כפי שניתן לשמוע הצלחנו להפריד את מקורות השמע באופן מוצלח יחסית.

כינור וחליל

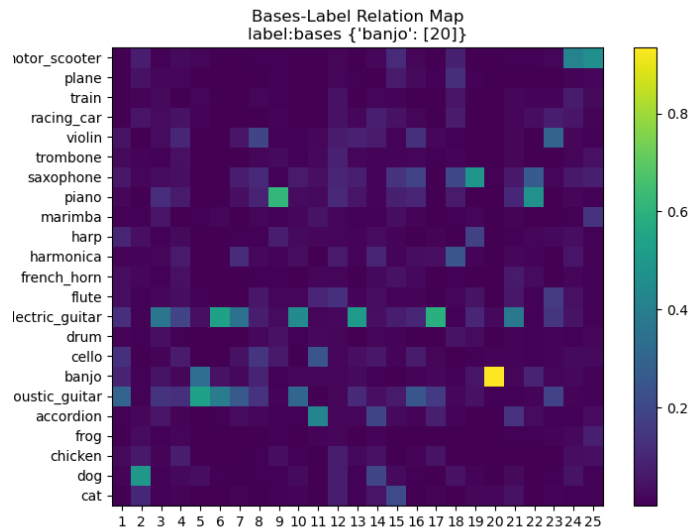


איור 42: ימין – מפת יחס בסיס – אובייקט, שמאל – פריים מהסרטון

כפי שניתן לראות מהאיור לעיל, המפה מציגה שגם ה – cello מהווה כמקור קול בסרטון וזאת מאחר שהאקטיביות שלו במפה זו יחסית גבוהות, אולם נשים לב שאנו לוקחים את המחלקות שה – ResNet חזה ולכן נבחר רק בשני לייבלים אלו (חליל וכינור). נקבל את ההפרדה הבאה:

- 
 sound_original.wav • שמע מקורי:
- 
 sound_violin.wav • שמע הכינור:
- 
 sound_flute.wav • שמע החליל:

Banjo – גיטרה אקוסטית 1



איור 43: ימין – מפת יחס בסיס – אובייקט, שמאל – פריים מהסרטון

כפי שניתן לראות מאיור 43, רשת ה- ResNet זיהתה רק את ה- Banjo ולא את הגיטרה האקוסטית, אולם הרשת שלנו כן הצליחה "להבין" שבשמע קיים גם גיטרה אקוסטית למרות שה- ResNet לא הצליח. דוגמה זו מבליטה את היתרון של לימוד צלילים של אובייקטים מאלפי סרטונים ללא תיוג; השיטה למדה את בסיסי השמע הנכונים לגיטרה, ו"שומעת" את נוכחותו בסרטון למרות שגיטרה איננה זוהתה ב- ResNet. דוגמה זו מצביעה על כך שרשת ה- MIML למדה בהצלחה את דפוסי הספקטרום של צלילים שונים, ומסוגלת לשייך בסיסי שמע לקטגוריות אובייקט.

במטרה לשפר את הרשת נוכל לבצע מס' דברים:

- מאחר שחלק מהשגיאות ברשת מתקבלות מחיזוי שגוי של ה- ResNet נוכל לשפר זאת אם נאמן את הרשת הזו עם הדאטא שאנו מאמנים את רשת ה- MIML ולא ע"י ImageNet, כך נקבל שחיזוי האובייקטים הנמצאים בפריימיים מדויק יותר.
- בחירה חכמה של הוקטורי בסיס עבור לייבל נתון – אנו השתמשנו בפרויקט זה בפעולת מקסימום בשביל לחשב את הסף שעבורו נשייך וקטורי בסיס ללייבל מסוים.
- מאחר וזיהינו שישנה בעיית overfitting ניתן לבחון הוספת שכבות dropout שיעזרו בבעיה זו (כפי שהסברנו במבוא).
- בחירה חכמה ושקלול נכון של מוצא רשת ה- ResNet עם הלייבלים שנותנים אקטיבציות גבוהות, כלומר במידה וקיבלנו בחיזוי הרשת של לייבלים מסוימים נכונים ובמפת בסיס – אובייקט אנו נוכחים לדעת שיש אקטיבציות גבוהות גם ללייבלים אחרים אזי נמליץ לבצע שקלול חכם שייקח את השני הדברים הללו בחשבון.



בסך הכל, התוצאות מבטיחות ומהוות צעד בולט לקראת הפרדת מקור שמע מונחה חזותית לסרטונים. כמובן שהמערכת שהוצגה רחוקה מלהיות מושלמת. מצבי הכשל הנפוצים ביותר בשיטתנו הם כאשר מאפייני השמע של אובייקטים שזוהו דומים מדי או שאובייקטים מזוהים באופן שגוי. יתר על כן, לא כל האובייקטים משמיעים צלילים ולא כל הצלילים נמצאים בטווח המצלמה

בפרויקט זה הצגנו מערכת ללימוד צלילי אובייקט מאלפי סרטונים לא מתויגים. רשת ה – MIML מקשרת את בסיסי השמע לאובייקטים המתאימים. שימוש בפירוק NMF לשני מטריצות אי – שליליות גרמה להצלחה בהפרדת הצלילים ברמת האובייקט. הפרדה בין מקורות השמע יכולה לשמש ליישומים רבים, למשל, מיון אירועי שמע וכדומה.

Bibliography

1. Gradient descent: <https://ruder.io/optimizing-gradient-descent/>
2. Adam: <https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c>
3. Optimizers: <https://mlfromscratch.com/optimizers-explained/#/>
4. SGD with momentum: <https://towardsdatascience.com/stochastic-gradient-descent-with-momentum-a84097641a5d>
5. Gradient descent and backprop: <https://towardsdatascience.com/a-step-by-step-implementation-of-gradient-descent-and-backpropagation-d58bda486110>
6. DNN: <https://www.oreilly.com/library/view/tensorflow-for-deep/9781491980446/ch04.html>
7. FC in CNN: <https://cs231n.github.io/convolutional-networks/#fc>
8. CNN: <https://towardsdatascience.com/convolutional-neural-network-17fb77e76c05>
9. FC: <https://iq.opengenus.org/fully-connected-layer/>
10. AlexNet: <https://arstechnica.com/science/2018/12/how-computers-got-shockingly-good-at-recognizing-images/3/>
11. DNN BN and Dropout: <https://towardsdatascience.com/training-deep-neural-networks-9fdb1964b964>
12. Train NN: https://ml4a.github.io/ml4a/how_neural_networks_are_trained/
13. Loss: <https://deeplearningdemystified.com/article/fdl-3>
14. Learning rate: <https://www.jeremyjordan.me/nn-learning-rate/>
15. ResNet: <https://cv-tricks.com/keras/understand-implement-resnets/>
16. Rahne, T., Bockmann, M., von Specht, H., Sussman, E.S.: Visual cues can modulate integration and segregation of objects in auditory scene analysis. *Brain research* (2007): <https://europepmc.org/article/med/17306232>
17. Jhuo, I.H., Ye, G., Gao, S., Liu, D., Jiang, Y.G., Lee, D., Chang, S.F.: Discovering joint audio-visual codewords for video event detection. *Machine vision and applications* (2014):
 - a. <https://link.springer.com/article/10.1007/s00138-013-0567-0>
18. Naphade, M., Smith, J.R., Tesic, J., Chang, S.F., Hsu, W., Kennedy, L., Hauptmann, A., Curtis, J.: Large-scale concept ontology for multimedia. *IEEE multimedia* (2006):
 - a. <https://ieeexplore.ieee.org/document/1667983>
19. F´evotte, C., Bertin, N., Durrieu, J.L.: Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural computation* (2009):
 - a. <https://ieeexplore.ieee.org/document/6797100>
20. Hyvarinen, A., Oja, E.: Independent component analysis: algorithms and applications. *Neural networks* (2000): <https://www.sciencedirect.com/science/article/abs/pii/S0893608000000265>
21. Barzelay, Z., Schechner, Y.Y.: Harmony in motion. In: *CVPR* (2007): <https://ieeexplore.ieee.org/document/4270342>
22. Casanovas, A.L., Monaci, G., Vandergheynst, P., Gribonval, R.: Blind audiovisual source separation based on sparse redundant representations. *IEEE Transactions on Multimedia* (2010): <https://ieeexplore.ieee.org/document/5466231?arnumber=5466231>
23. Arandjelovic, R., Zisserman, A.: Look, listen and learn. In: *ICCV* (2017): <https://ieeexplore.ieee.org/document/8237335>
24. Arandjelović, R., Zisserman, A.: Objects that sound (2017): <https://arxiv.org/abs/1712.06651>