Part 1: CIFAR10 2.0 Electric Boogaloo

חלק זה של המטלה מהווה המשך ישיר למטלה הקודמת. מטרת החלק הייתה לבנות, לאמן ולהעריך רשת קונבולוציה פשוטה לבעיית סיווג של תמונות ממערך הנתונים CIFAR10. נזכיר שמדובר במערך של 60,000 תמונות קטנות (32*32 פיקסלים, בשלושה ערוצים: אדום, ירוק וכחול) אשר כולן מתויגות ומחולקות ל-10 קטגוריות (מטוס, כלב, וכו). בכתיבת הקוד, לקחנו את הבסיס של Fully Connected Networkשיצרנו במטלה הקודמת ובנינו ממנו רשת נוירונים קונבולוציונית (CNN) במטרה לפתור את מטלת CIFAR10 באופן מלא. היעד הינו להבין את השפעת ארכיטקטורת הרשת, תהליך האימון ופרמטרים כמו norm על היכולת של המודל ללמוד להבחין בין קטגוריות שונות וכמובן להגיע לרמת דיוק של מעל 85%. עם מודל הקונבולוציה התוצאות הכי טובות שלנו היו 84.7% אחוזי דיוק, ולכן לקחנו את המודל הכי טוב שהצלחנו לבנות (בעיקר את ההיפר-פרמטרים הנוגעים למספר הערוצים של שכבות הקונבולוציה) והימרנו אותו לגרסא שלנו של ResNet כלומר הוספנו חיבורים שיוריים (Residual/Skip connections) אשר גרמו לנו להגיע ליעד שלנו ואף לחצות אותו.

<u>ארכיטקטורת המודל</u>

מודל ה-CNN הראשוני הכי טוב שהצלחנו לבנות היה בעל ארבע שכבות קונבולוציה אשר כל אחת מהן כוללות Batchnorm, ReLU, MaxPool, Dropout וארבע שכבות Fully Connected. הדבר הראשון שדאגנו להכניס בעקבות מה שלמדנו בהרצאות, היה נירמול של הקלט של המודל על ידי חישוב ערכי ה-mean וה-std של סט האימון ושימוש במידע זה בכדי לנרמל את כל התמונות בכל הסטים. דבר נוסף שעזר לשפר את הביצועים שלנו היה לשחק הרבה עם גודל הקרנל של כל שכבה וזה הביא לשיפור גדול בדיוק (התוצאות הכי טובות שיצאו לנו עם הCNN היו ארבע קרנלים בגדלים הבאים: 7,5,3,2 הביאו אותנו עם הנורמליזציה של התמונות ל 84.7%).
אך לבסוף החלטנו לעבור לרשת ResNet, ללמוד כמה שיותר מן ההיפר-פרמטרים שהביאו לתוצאות הכי טובות, לפשט את הכל ולהתחיל את החקירה שלנו מחדש. זמן הריצה של הרשת הראשונה שהקמנו עם חיבורים שיוריים רצה זמן ארוך מהמצופה אז החלטנו להחזיר את גדלי הקרנלים לערכים קטנים ואז הרשת החדשה עלתה על ציפיותנו ונשארנו איתה (ארבע קרנלים בגודל של 2).
ההבדלים המשמעותיים בארכיטקטורה של הרשת החדשה היא: במקום רצף ישר של שכבות קונבולוציה ו-Fully Connected, רשת ה-ResNet שלנו בנויה מבלוקים שיוריים (Residual Blocks), שבה כל בלוק כזה מורכב משתי שכבות קונבולוציה, Batch Normalization ו-ReLU. החיבורים השיוריים מאפשרים למידע לדלג על שתי שכבות קונבולוציה להתווסף ישירות לפלט של הבלוק. גישה זו עוזרת להתמודד עם בעיית היעלמות הגרדיאנט ומאפשרת אימון של רשתות עמוקות יותר.

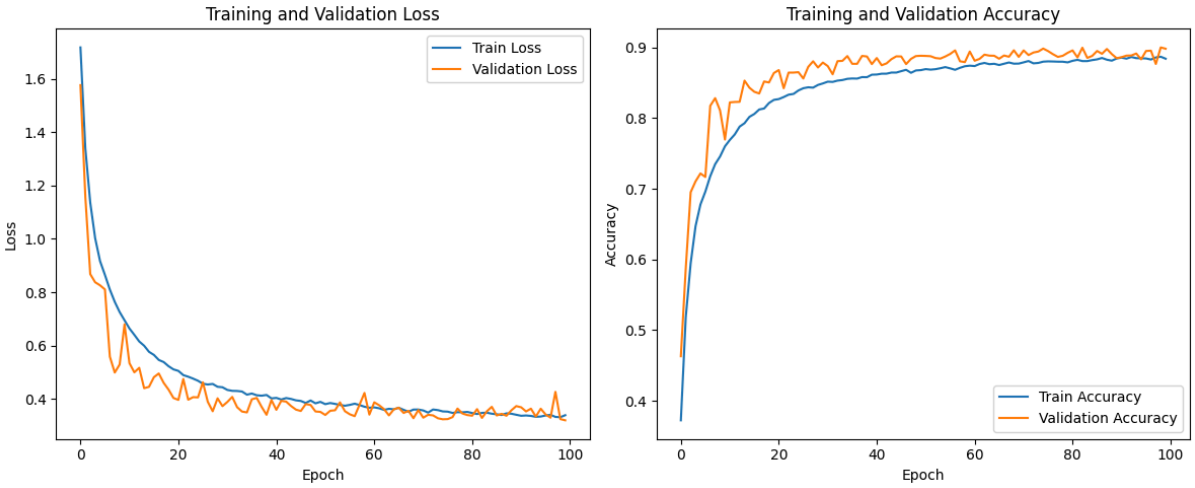
הרשת הסופית שהשתמשנו בה מורכבת מ-4 שכבות כאלו, כאשר כל שכבה כזו מכילה מספר בלוקים שיוריים. מספר הערוצים עולה פי 2 בין השכבות, החל מ-64 בשכבה הראשונה ועד ל-512 בשכבה הרביעית. לפני החיבור השיורי הראשוני, קיימת שכבת קונבולוציה ראשונית (self.conv1) עם Batch Normalization ו-ReLU. בסוף הרשת, במקום מספר שכבות Fully Connected, יש שכבת Pool ממוצע גלובלי (F.avg_pool2d) שמקטינה את המאפיינים לשכבה לינארית יחידה (self.linear) המפיקה את 10 הסיווגים הסופיים.

השימוש בבלוקים שיוריים במבנה מודולרי זה הקל על אימון הרשת העמוקה יותר ואפשר להשיג ביצועים טובים יותר בהשוואה למודל ה-CNN הראשוני.

אימון המודל

optimizer: NAdam
Loss: Cross Entropy Loss
Batch Size: 128
Epochs: 128

תוצאות




```
Epoch 1/100 - Train Acc: 0.3723, Val Acc: 0.4630 | Train Loss: 1.7184, Val Loss: 1.5769
...
Epoch 30/100 - Train Acc: 0.8492, Val Acc: 0.8788 | Train Loss: 0.4434, Val Loss: 0.3726
...
Epoch 60/100 - Train Acc: 0.8744, Val Acc: 0.8944 | Train Loss: 0.3666, Val Loss: 0.3408
...
Epoch 90/100 - Train Acc: 0.8845, Val Acc: 0.8850 | Train Loss: 0.3402, Val Loss: 0.3727
...

Epoch 100/100 - Train Acc: 0.8842, Val Acc: 0.8984 | Train Loss: 0.3386, Val Loss: 0.3194
```

Test Loss: 0.3675, Test Accuracy: 0.8919

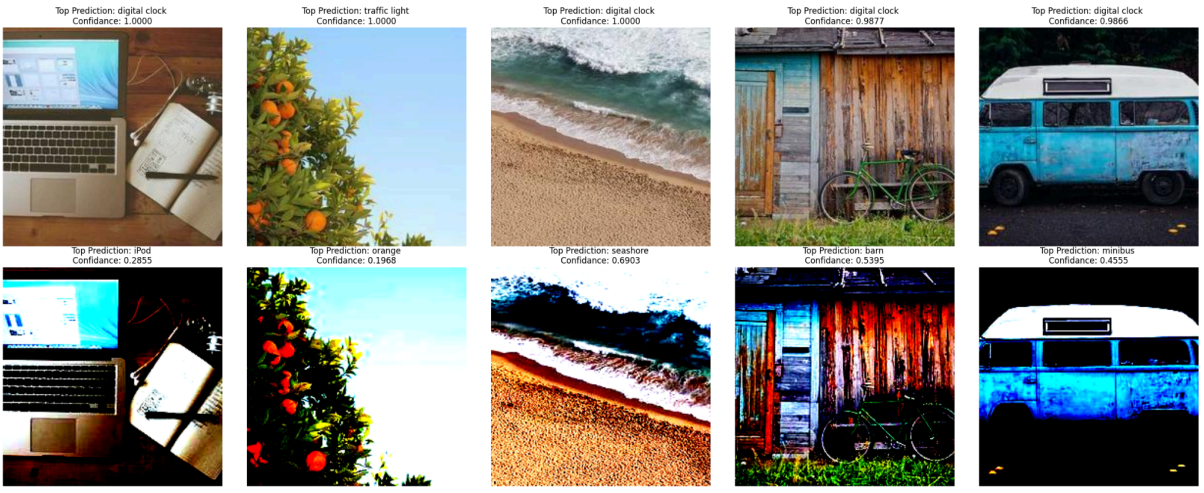
מהגרפים ניתן לראות שהמודל המשופר הציג ביצועי סיווג טובים משמעותית: לאחר 100 אפוקים התקבלו דיוק אימון של 88.4%, דיוק ולידציה של 89.8% ודיוק על קבוצת הבדיקה של 89.2%. בשני הגרפים רואים ירידה יציבה ב-Loss, עליה מתמדת בAcc, והתכנסות טובה בין קבוצות הנתונים, כך שניתן להסיק שהמודל הצליח בלמידה כללית ונמנע מ overfitting. התוצאה מעידה כי המודל המשופר מתאים יותר לבעיה ולמערך הנתונים של CIFAR-10.

בגרף השמאלי ניתן לראות ש־Train Loss יורד באופן יציב לאורך האפוקים ומתכנס לערך נמוך מאוד (כ־0.33). גם Validation Loss מציג ירידה יפה, עם רעש מתון. העובדה ששני הגרפים שומרים על מגמה דומה וללא פער גדול מצביעה על למידה אפקטיבית והכללה טובה של המודל – אין חשש ל־overfitting.

בגרף הימני ניתן לראות ש־Train Accuracy עלה בצורה חדה, והגיע עד לרמה של כ־88.5%. ה־Validation Accuracy אף גבוה יותר — עד 90%, עם תנודתיות קטנות בלבד. זהו סימן לכך שהמודל לא רק "זוכר" את הדאטה, אלא גם לומד תבניות כלליות. הפער בין train ל־val קטן מאוד, במיוחד בשלבים המאוחרים – ניתן להסיק שקיים איזון נכון בין למידת יתר לחוסר למידה.

Part 2: Activation Maximization

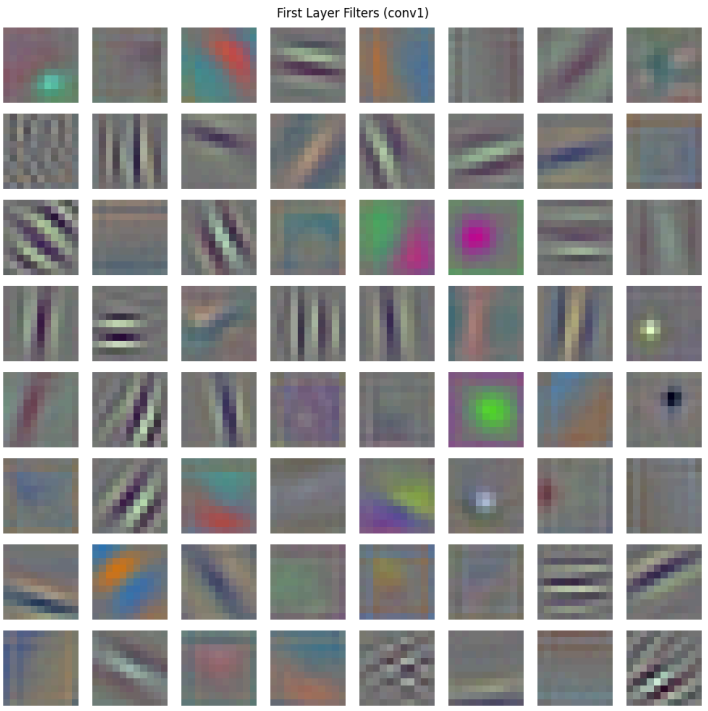
1a) בחלק השני של העבודה, יישמנו activation maximization על רשת AlexNet, בניסיון להבין טוב יותר את הפונקציונליות של שכבות שונות במודל קונבולוציוני מאומן מראש.
בשלב הראשון נרצה לוודא שטענו את AlexNet כראוי. בחרנו חמש תמונות אשר מכילות אובייקטים שנמצאים באחד מן הקטגוריות של alexnet, והרצנו עליהן שני ניבויים - פעם אחת ללא נירמול ובפעם השנייה עם נרמול לפי סטטיסטיקות של ImageNet (כלומר mean ו- std) והשווינו את התוצאות שקיבלנו. את התמונות הבאנו מן אתר חופשי לתמונות ברשת Lorem Picsum ואת הנתונים של הנירמול הבאנו מן האתר הרשמי של torch שמראה כיצד ליישם את alexnet בקוד פייתון.



1b) הניתוח העלה הבדלים משמעותיים במדדי הבטחון של המודל. ניתן לראות כי ללא נירמול המודל הפיק תחזיות אקראיות עם ביטחון נמוך, בעוד שנירמול שיפר משמעותית את רמת הביטחון. בדוגמאות לניתוח ללא נירמול, ברוב המקרים התחזית נעה בין 0.19 ל־0.53, ולעיתים אף פחות. התחזיות היו לא עקביות, ובחלק מהמקרים לא תאמו כלל את תוכן התמונה. לדוגמה, התמונה של עץ התפוזים סווגה בטעות כ־traffic light עם ביטחון של 100%. לעומת זאת, לאחר ביצוע נירמול לפי סטטיסטיקות ImageNet, חלה קפיצה דרמטית בביטחון המודל: ארבע מתוך חמש התחזיות הגיעו לרמת בטחון של מעל 98%, מה שמעיד על תגובה חזקה של הרשת לקלט שעבר התאמה לפורמט שהיא "רגילה" אליו. יחד עם זאת, חשוב לציין כי גם כאשר הביטחון היה גבוה מאוד —

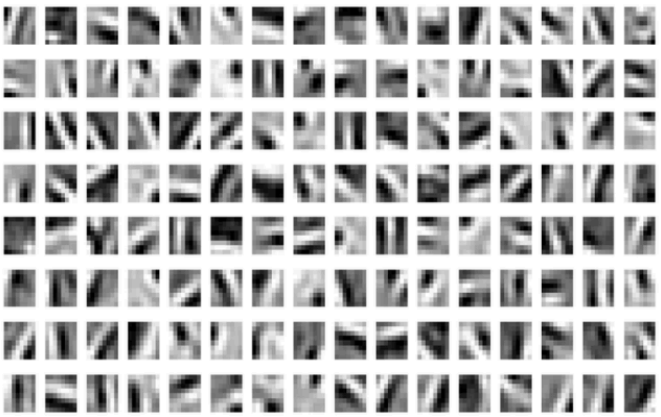
התחזיות לא תמיד היו נכונות. לדוגמא, המחשב והמחברת סווגו כ״אייפוד״. כלומר, הנירמול שיפר את הוודאות של המודל, אך לא בהכרח את הדיוק. התוצאות ממחישות את התלות של המודל בפרה־פרוססינג, ואת הרגישות שלו למאפיינים שטחיים כמו טקסטורה, צבע וניגודיות – יותר מאשר לזיהוי אובייקטים שלמים.

2a) בשלב השני התבקשנו להציג ויזואליזציה של פילטרים בשכבה הראשונה. לצורך ביצוע המשימה, שלפנו את משקלי שכבת הקונבולוציה הראשונה (conv1) במודל ונירמלנו את הערכים לתחום בין 0 ל1 והצגנו את 64 הפילטרים במערך של 8*8 RGB.



מהסתכלות על הפילטרים אנחנו רואים תבניות של קווים באוריינטציות שונות עם אזורים צבעוניים וניגודיות גבוהה.

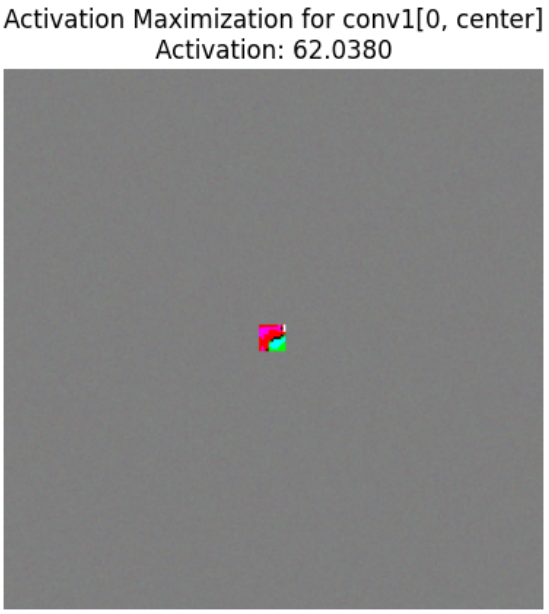
2b) יש דמיון גדול בין הפילטרים לנוירונים בV1 (קורטקס ראייתי ראשוני) שמגיבים לקווים, צבעים ותדרים מרחביים, ניתן לראות זאת במאמרים שנעשו בנושא (Jehee and Ballard 2009) שבהם הם מציגים את הצורות הפשוטות שנמצא כי הנוירונים הראשוניים בV1 מגיבים הכי חזק אליהם:



2c) השכבה השניה בalexnet מקבלת קלט עם 64 ערוצים ומוציא פלט של 192 ערוצים עם קרנל של 5 על 5.
2d) מספר הערוצים בשכבה השנייה הוא גדול בהרבה משלושת ערוצי הצבע שאנחנו בני האדם יודעים לתפוס בעינינו (לעומת השכבה הראשונה) ולכן לא ניתן להציג אותם כתמונה רגילה.

3a+b) בשלב זה יצרנו תמונה בגודל 3*224*224 כאשר ערכי הפיקסלים נדגמו מהתפלגות נורמלית עם ממוצע 0.5 וסטיית תקן 0.01. לאחר מכן, השתמשו בFeatureExtractor כדי לבנות מודל שבו הפלט הוא השכבה הקונבולוציונית הראשונה של alexnet לפני הפעלת ReLU & BatchNorm. יעד האקטיבציה (optimization objective) הוא היחידה במרכז הערוץ הראשון של השכבה, כלומר conv1[0, center, center] וביצענו עליו Gradient Ascent. בכל איטרציה חישבנו את הגרדיאנטים של הפלט ביחס לתמונה ועדכנו את ערכי הפיקסלים בכיוון הגרדיאנט עם Step Size אלפא 5.0 (לפי ההמלצה a>1). לאחר כל עדכון, תחמנו את ערכי הפיקסלים כך שישארו בטווח [0,1] באמצעות img.clamp_(0,1).

לאחר 30 צעדים קיבלנו את התמונה הבאה:

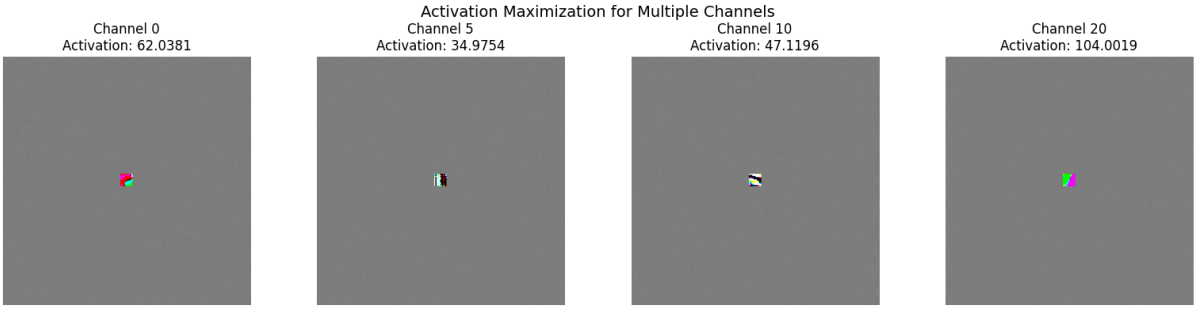


במרכז התמונה מופיעה תבנית צבעונית בולטת בגווני ירוק ורוד ולבן ושאר הפיקסלים נותרו בגוונים אפורים ללא שינוי משמעותי, מכיוון שהאקטיבציה מבוססת על פיקסל אחד (יחידה מרכזית אחת). כלומר, התמונה משקפת את ההעדפה המקומית של הפילטר הראשון של conv1 - ככל הנראה, הפילטר מגיב לתבניות צבעוניות חדות וקטנות. הפלט מראה כיצד רשתות קונבולוציה לומדות לזהות תכונות בסיסיות כבר בשכבות הראשונות – כמו צבעים וניגודיות. ערך האקטיבציה שקיבלנו - 62.0380 - מייצג את עוצמת התגובה של הפילטר לתמונה שיצרנו. כלומר, הערך גבוה ומעיד על כך שהתמונה שנוצרה מכילה את התכונה הויזואלית שהפילטר רלוונטי עבורו, משמע, התמונה הצליחה לבטא תכונה כלשהי (יחסית בסיסית) שהרשת כבר למדה לזהות, גם אם עבורנו (עין אנושית) אין שום משמעות לקוביה הזאת

אכן יש קשרים ויזואליים בין התמונה לבין הפילטרים משלב 2. גם בפילטרים וגם בתמונה יש צבעים חזקים וניגודיות גבוהה בצבעי בסיס (אדום, ירוק, כחול). בנוסף, יש חזרתיות של תמונה מתקבלת עם תבנית מאוד מקומית ומאוד מרוכזת - כלומר, פילטר שמחפש מאפיינים חזותיים חדים באזורים קטנים. מה שכן, חשוב לזכור שהפילטרים עצמם הם משקלים, כלומר, הם בודקים את הקלט - האם קיים כאן קו/ צבע וכו׳. לעומת זאת, התמונה שקיבלנו היא בעצמה קלט לפילטר והיא נראית יותר מלאכותית. בנוסף, לפילטרים יש תבניות סימטריות והתמונה שקיבלנו היא בעיקר צבעים חדים ללא צורה מוגדרת.

3c+d) בשלב זה חזרנו על התהליך מחלק b עבור ערוצים נוספים בשכבת conv1. רצינו לבדוק כיצד משתנות התמונות המתקבלות כתוצאה ממיקסום האקטיבציה של יחידות שונות, והאם ניתן להסיק על ה"תכונה" שכל פילטר מזהה.

בדקנו ארבעה ערוצים - 0, 5, 10, 20. כמו בסעיף הקודם התחלנו ביצירת התמונה הרועשת עם ממוצע 0.5 וסטיית תקן 0.01 וביצענו 30 צעדים של Gradient Ascent עם סטפ סייז 5.0 ווידאנו שהתמונה נשארה בטווח [0,1].

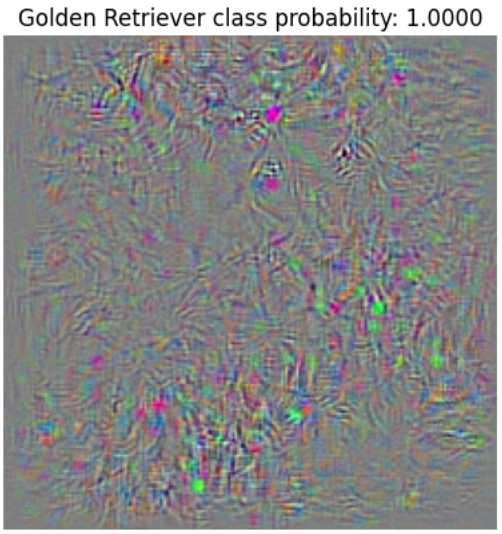


קיבלנו ארבע תמונות שונות כאשר ערוץ 0 וערוץ 20 יחסית דומים מבחינת צבעים בוהקים וערוצים 5 ו-10 מראים דפוסים קצת שונים עם צבעים וצורות שונות.

ערכי האקטיבציה שהתקבלו עבור כל אחד מהערוצים משקפים את עוצמת התגובה של הפילטר לתמונה שנוצרה במיוחד עבורו. ערוץ 20 הוא בעל ערך האקטיבציה הגבוה ביותר - 104 וערוץ 5 בעל הערך הנמוך ביותר שהתקבל - 34.98. אנחנו מסיקים שהפערים נובעים מכך שהתמונה שנוצרה עבור ערוץ 20 מתאימה במידה הגבוהה ביותר לדפוס שהפילטר למד לזהות - איזושהי תכונה חזותית שהרשת למדה כחשובה בשלב מוקדם יותר באימון. לעומת זאת, הפילטר בערוץ 5 הופעל בעוצמה נמוכה יחסית לשאר. אפשר להסיק שהתכונה שהפילטר מזהה כנראה מורכבת מאוד או פחות חד משמעית. חלק זה מדגיש את ההבדל במידת הספציפיות והרגישות של הפילטרים השונים לשינויים בתמונה. בנוסף, אפשר לראות את הפוטנציאל של שימוש באקטיבציה להבנת תהליך הלמידה שהרשת עברה - איזו תבניות היא ״מעדיפה״ ומזהה בקלות.

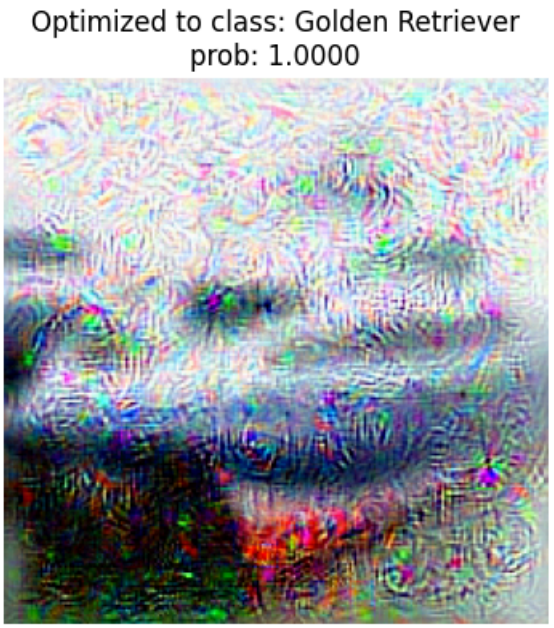
4) בשלב זה ניסינו ליצור תמונה סינטית שגורמת למודל לסווג אותה כשייכת למחלקה מסוימת - במקרה הזה מחלקה 207 - גולדן רטריבר, באמצעות מקסום ההסתברות של אותה מחלקה אחרי Softmax. כמו בשלבים קודמים, התחלנו מתמונה רועשת (noise) בגודל 3*224*224, עם ממוצע 0.5 וסטיית תקן 0.01. ביצענו 40 צעדים של Gradient Ascent על הסתברות המחלקה ורצינו לראות האם המודל יזהה אותה ככלב מסוג גולדן רטריבר.

התקבלה תמונה עם טקסטורה לא טבעית ותבניות צבעוניות רנדומליות. בסופו של דבר ההסתברות שהתקבלה הייתה 1. כלומר, המודל בטוח ב100% שהטקסטורה הצבעונית הזו הינה גולדן רטריבר.



למרות שעבור העין האנושית מדובר בתמונה מופשטת ורנדומלית, היא ככל הנראה מכילה תבניות שמפעילות פילטרים רבים הקשורים לזיהוי כלבים במודל. באימון, הרשת למדה לזהות תכונות כמו פרווה, זנב, צבעים וכו׳ – והתמונה שנוצרה מצליחה לעורר את התגובה המשולבת של פילטרים אלה ולשטות במודל.
העובדה שהגענו להסתברות של 100% מצביעה על כך שהאופטימיזציה הייתה מאוד אפקטיבית – גם בלי מידע חזותי הגיוני לאדם, הרשת קובעת בוודאות מוחלטת שמדובר בגולדן רטריבר.
התוצאה מדגימה כיצד ניתן ליצור בעזרת גרדיאנטים תמונה שגורמת לרשת לזהות קטגוריה מסוימת, גם כאשר אין אובייקט אמיתי או הגיוני בתמונה.

5) בשלב האחרון לקחנו תמונה של ספינה מתוך התמונות של cifar10 והגדלנו אותה לרזולוציה של 224*224 המתאימה ל-alexnet ובחרנו את מחלקת היעד להיות גולדן רטריבר, וביצעו 40 צעדים של Gradient Ascent על ההסתברות לאחר סופטמקס.



התמונה שונתה רק בעדכונים קטנים והדרגתיים כך שעין אנושית שחשופה גם לתמונה המקורית יכולה לראות (אולי לא באותה קלות כמו בהתחלה) שעדיין מדובר בספינה. ההסתברות הסופית שהתקבלה הייתה 1 – זאת אומרת שהמודל בטוח ב-100% שהתמונה מייצגת גולדן. כלומר, די בשינויים הקטנים על מנת לשטות במודל ולגרום לו לשנות לחלוטין את הסיווג שלו.